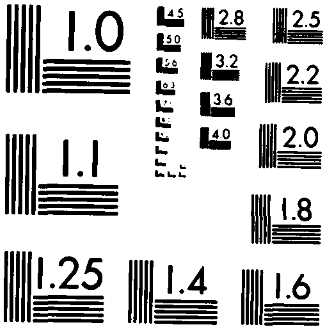
AD-A189 744 DEVELOPING NEW USER INTERFACES FOR THE THEATER WAR 1/1
EXERCISE(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB
OH SCHOOL OF ENGINEERING M S KROSS DEC 87
UNCLASSIFIED AFIT/GCS/ENG/87D-19 F/G 15/6 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(1)

DEVELOPING NEW USER INTERFACES FOR THE THEATER WAR EXERCISE

THESIS

Mark Scott Kross
Captain, USAF

AFIT/GCS/ENG/87D-19

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/87D-19

DEVELOPING NEW USER INTERFACES FOR THE THEATER WAR EXERCISE

THESIS

Mark Scott Kross
Captain, USAF

AFIT/GCS/ENG/87D-19

Approved for public release; distribution unlimited

AFIT/GCS/ENG/87-19

DEVELOPING NEW USER INTERFACES FOR THE THEATER WAR EXERCISE

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science (Computer Systems)

Mark Scott Kross, B.A.
Captain, USAF

December, 1987

Approved for public release; distribution unlimited

Preface

The goal of this thesis was to design and implement new user interfaces for the Theater War Exercise (TWX). This thesis was one half of a two thesis effort which included designing a new database system for TWX and rewriting the TWX air and land battle simulation programs. This effort included rehosting the TWX user interfaces and other software from a mainframe to a microcomputer environment using the Ingres application development tool.

The Theater War Exercise (TWX) is a two-sided, theater-level, computer-assisted wargame played at the U.S. Air Force Air War College and supported by the Air Force Wargaming Center. It is also played at the Air Force's Combined Air Warfare Cource (CAWC), and at intermediate service schools in both Canada and England. It is an exercise in force posturing and employment at the Central European theater level.

This thesis presents background on TWX, current literature on the process of building user interfaces, and an approach to building the TWX user interfaces which eliminates many of the problems associated with the old TWX system.

I wish to thank my thesis advisor, Captain Mark Roth, for his tremendous assistance and patience during this thesis effort. I also wish to thank my thesis partner, Captain Michael Brooks, for his encouragement, dedication, and hard work. His positive attitude made a tough project much easier to work on. I wish to thank Captain James Jansen, who helped me immensely in learning to use the Ingres development environment. I would also like to thank Dean J. Przemieniecki, Colonel James Weaver, and Major Dan Reyen for their support as thesis committee members and readers. Additionally, I wish to thank the Air Force Wargaming Center for their support of this thesis effort. Finally, and most importantly, I wish to thank my wife Maria, to whom I was married less than a week before coming to AFIT. Her support throughout my time at AFIT was immeasurable.

Mark Scott Kross

Table of Contents

List of Figures

AFIT/GCS/ENG/87-19

Abstract

'The Theater War Exercise (TWX) is a five-day, two-sided, theater level, computer assisted wargaming exercise. TWX is an exercise in theater-level airpower employment decision making. It is played at the Air War College and at intermediate service schools in England and Canada. It is maintained by the Air Force Wargaming Center.

The TWX software consists of land and air battle simulations and user interface programs. The user interfaces include both input and output programs.

There were two problems with the current TWX system that needed to be corrected. The first was that due to its mainframe application, TWX was costly to use for the Canadian and British schools. The second problem was that all input and output for TWX was made over hard copy devices, making TWX difficult and frustrating to use for exercise players.

To solve the problems with TWX, this thesis effort consisted of moving the entire exercise to a microcomputer configuration and designing and implementing new, screen-oriented user interfaces emphasizing ease of use and flexibility.

First, hardware and software for the development system was selected. Decisions in the hardware and software selection area were affected by the wishes of the Air Force Wargaming Center.

The body of literature concerning user interface design was then reviewed. From this literature, a design methodology to be used for the TWX user interfaces was developed.

The user interfaces themselves were then built. Certain aspects of the development effort proved especially interesting. Among these aspects were locking considerations, use of the selected software tools, and programming problems which arose during development.

Finally, areas for further research and future enhancement of the TWX system were explored. Some of these areas are of interest to those in the field of strategic and tactical sciences, as well as those interested in software design.

DEVELOPING NEW USER INTERFACES FOR THE THEATER WAR EXERCISE

I. Background and Problem Definition

Background of the Theater War Exercise

The Theater War Exercise (TWX) is currently a five-day, two-sided, theater level, computer assisted wargaming exercise. TWX is an exercise in airpower employment decision making. The decisions required of exercise participants are typical of those that an air component commander and staff would make during an actual war. These decisions, once made by the exercise participants, are fed into TWX's air and land battle simulation programs. These programs simulate the employment of airpower doctrine, strategy, and warfighting principles based on participant input. The players receive the battle results, air/land orders of battle, logistics status, weather forecasts, and other information in a computer run format.

Currently, TWX is an in house exercise run at the Air Force Wargaming Center, Maxwell Air Force Base, Alabama. The exercise runs on a Honeywell H6000 series computer, with inputs made on TI Silent 700 series terminals. All input and output is done on hard copy devices with limited and slow interaction by exercise controllers and participants. TWX is also played each year at the Canadian Forces Command and Staff College (CFCSC) and at the Royal Air Force Staff College. At these sites, computing facilities similar to those at Maxwell AFB are leased and Air University personnel load and operate the exercise.

The two sides which are played in TWX are NATO Blue and Warsaw Pact Red. A single group of players is referred to as a seminar group. Each side does its logistics and mission planning once each day. The plans created are then input into the computer. The TWX simulation programs then use this input to determine the outcome of the day's battle. The status of the conflict is then output by the TWX output programs and distributed to game players the next day. Typically, the simulation programs are executed during the night.

The TWX software consists of the land and air battle simulation and the user interfaces. The user interface includes both the input and output programs. Currently, the user interface allows control over aspects of the air battle only, as the land battle is programmed to run in a predetermined manner by Wargaming Center personnel before the first day of the exercise. The only interactions between the land and air battles takes the form of close air support and battlefield air interdiction missions flown against land targets by aircraft which are part of the air battle.

Problem Definition

The exercise of warfighting skills is a critical part of the education process of Air Force officers. As a result, the Air Force Wargaming Center desires to have TWX widely available. TWX should also be as easy to use as possible.

There are two problems with the current TWX system that need to be corrected. The first is that due to its mainframe application, TWX is costly to use for the Canadian Forces Command and Staff College and the Royal Air Force Staff College. The cost of the mainframe application also makes TWX unavailable to many Air Force officers who could derive great benefit from participating in the exercise.

The second problem is that the hard copy nature of TWX input and output makes it difficult and frustrating to use for the exercise players. The reason that hard copy is difficult to work with is that all inputs must be precisely formatted by hand before the participants ever interact with the TWX computer. Hard copy is frustrating to use because mistakes made during input cannot be corrected. A screen-oriented user interface of some type is needed to overcome this problem.

Currently, controller personnel are used to help exercise participants overcome some of the problems of TWX. These controllers are Air Force Wargaming Center personnel. Unfortunately, the controllers also suffer with the same problems associated with hard copy devices.

Proposed Solution

To solve the problems with TWX, this thesis effort will consist of the following tasks:

1. Moving the entire exercise to a configuration based on a combination of readily available microcomputers.
2. Designing and implementing new user interfaces with an emphasis on ease of use and flexibility.

In addition to the above steps, several other tasks must be accomplished in order to fully reconfigure TWX:

1. Design and implementation of a relational TWX database system to improve exercise flexibility and reduce data redundancy.
2. Rehosting of the TWX land and air simulation programs to the new machine.
3. Implementation of a controller interface which will give exercise controllers the power needed over TWX play along with the flexibility of the player user interfaces.

The above considerations were addressed in a separate thesis effort by Capt Mike Brooks [5].

Assumptions Made

Several assumptions are made for the purposes of this thesis effort. These assumptions originated due to either Wargaming Center direction or because of the nature of TWX itself. Among the areas affected by the assumptions made are hardware and software selection for the development effort, orientation of the user interfaces, and the rehosting of the TWX air and land battle simulations. Chapter 2 addresses the selection of development hardware and software and the inherent assumptions made.

The second assumption concerns the orientation of the new TWX user interfaces. They must be screen-oriented since using computer terminals is currently the only practical way to avoid the use of hard copy devices for input. By deciding to develop screen-oriented interfaces for TWX, the possibility of using forms generation tools to increase programming productivity could also be explored. This aspect of the development effort is discussed further in Chapter 2.

A final assumption concerns the rehosting of the TWX air and land battle simulations. During this effort, the equations and calculations which form the nucleus of the simulations must not be changed. This is because the Wargaming Center is currently satisfied with the simulations themselves. Only the portions of TWX dealing with the user interfaces and the database will change. This will require that the simulation equations themselves be separated from the portions of the simulations which are unique to the mainframe computer currently used and the version of FORTRAN that the simulations are written in.

Sequence of Presentation

First, the hardware and software selection considerations for the development system will be discussed. Decisions in the hardware and software selection area will be largely affected by the wishes of the Air Force Wargaming Center.

The body of literature concerning user interface design will be reviewed. From this literature, concepts for a design methodology to be used for the TWX user interfaces will be extracted and presented.

The user interfaces themselves will then be discussed along with specific aspects of the interface development. Among these aspects will be multi-user locking considerations, use of the selected software tools, and programming problems which arise during development. Testing of the user interfaces will also be discussed.

Finally, areas for further research and future enhancement of the TWX system will be presented. Some of these areas will be of interest to those in the field of strategic and tactical sciences as well as those interested in software design.

II. Hardware and Software Selection for Development

Hardware Selection

In selecting the hardware for the new TWX system there were several factors to consider. First, a multi-user system or a local-area-network (LAN) configuration was required in order to support a minimum of three users, ie., blue players, red players, and controller. Second, a powerful central processor was needed to run the actual game simulations in an acceptable amount of time. In most cases, the actual TWX simulations are run at night so turnaround time is not critical. However, the exercise schedule currently requires a 4 hour maximum turnaround for the final day of the exercise.

Finally, a large amount of secondary disk storage was required to store the TWX database and simulation programs. Depending on the number of seminars the system must support, anywhere from 20 to 100 MB of storage is needed.

To solve these requirements and still meet our goals of portability we decided to use a DEC MicroVAX II, Zenith Z-158 combination as depicted in Figure 1. By connecting the Z-158s and MicroVAX together, the MicroVAX can perform as a central file server for concurrent exercises, with the Z-158s running the user interface software. Also, the MicroVAX, using the MicroVMS operating system, has the capability to run batch jobs which would allow large reports to be run in the background, thus freeing up the individual player's terminal for other processing requirements.

Rationale for Using a DBMS

The database function within the current TWX system is handled by a collection of independent application programs. These programs access multiple data files which reflect an emphasis on physical implementation rather than logical data organization. Due to this database setup, a large amount of data redundancy exists and the data structures cannot be modified easily without extensive changes to the TWX system. Also, in order to re-host the TWX system, the current

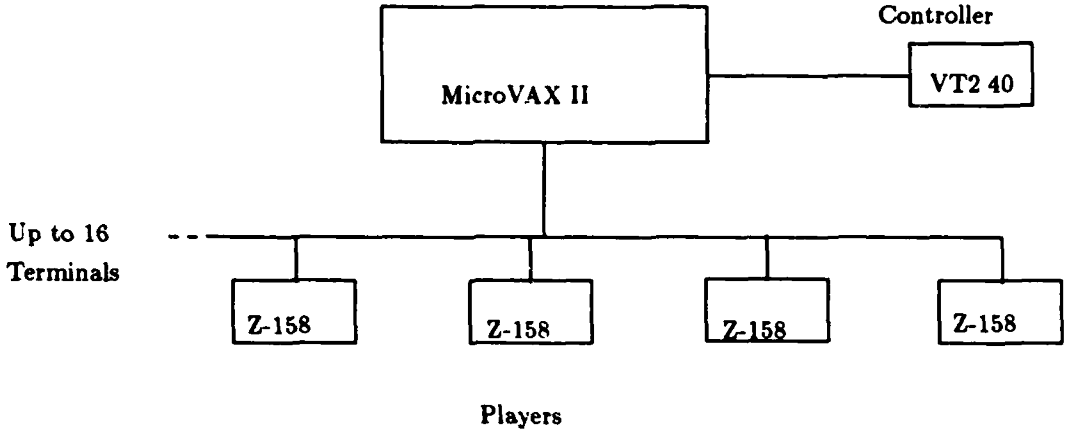


Figure 1. New TWX Hardware Configuration

specialized Fortran IV I/O application subroutines would have to be rewritten for the new microcomputer version. Since the current file system contains these drawbacks and recoding the current system would require excessive time and resources, it was deemed more advantageous to incorporate a vendor supplied DBMS in place of the current application database system [5].

As a result of choosing to use a vendor supplied DBMS, the I/O calls within the actual simulation programs were replaced by calls to the DBMS [5]. Also, all database updates from inputs into the user interface were made through the DBMS.

DBMS Selection

During the DBMS selection process, the choice of DBMS to use was narrowed down to three database management systems which met most of the TWX DBMS requirements. They were: MDBS III, an extended-network DBMS, Oracle, a relational DBMS, and Ingres, also a relational DBMS.

Overall, the best package with regard to the database functions for the new TWX implementation was MDBS III. MDBS III, using the extended-network model, was better suited to handle the intricate data relationships required for the new TWX database. However, at the time of this evaluation, the MDBS III PC version did not support a 4GL (Fourth Generation Language) development tool or a forms generator. Without these features, the processing power of the Z-158 could not be utilized unless the user interface software was developed in a higher-order language such as Pascal. Since a 4GL and forms generation system were needed in order to rapidly prototype the user interface, MDBS III was eliminated from consideration.

The selection process, therefore, focused on comparing Oracle and Ingres. Both database management systems were proven performers and each had a powerful 4GL application development tool which included a forms development package. Both systems provided all the capabilities necessary to develop the new user interfaces. Also, both systems have a powerful report writer and support the Structured Query Language (SQL) which is a standard query language used for relational databases. SQL was desirable for reasons which will be explained later.

The advantages of using a relational database system were many. First, relational databases, when properly designed, reduce data redundancy. Second, using a relational model made the database tables and the data in them easy to understand and relate to aspects of the TWX game. Finally, a substantial theory has developed regarding relational databases and the relational database model [11, page 45]. This theory aided in the actual design of the TWX database.

Although Oracle rated higher in the 4GL and forms management area and was our initial choice, Ingres was selected for the project due to more pragmatic considerations. Since the Air Force Wargaming Center is currently using Ingres for other in-house database applications and already has a trained support staff, Ingres was selected to preclude unnecessary training. This factor, plus significant cost reduction incentives, made Ingres the logical choice. Figure 2 depicts the proposed Ingres DBMS setup for TWX.

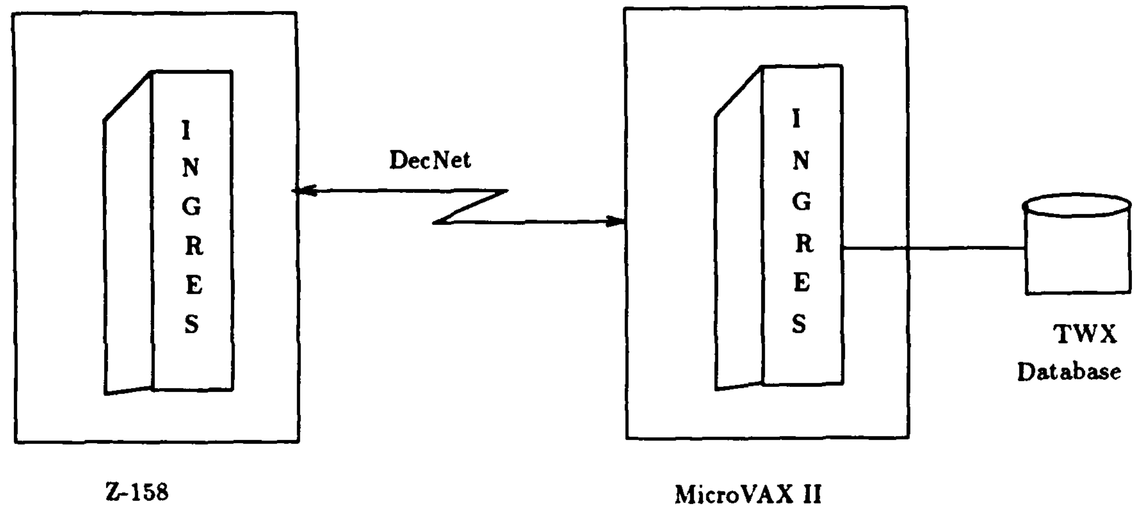


Figure 2. Proposed TWX DBMS Configuration

Selection of a Database Query Language

An interesting aspect of the design of the user interfaces was the selection of a database query language. The INGRES package supports two different query languages: SQL (Structured Query Language), and QUEL (an INGRES-specific query language). The decision was made to use SQL since it is becor 'r ; the de facto standard in database retrieval languages. This selection will enhance the maintainability of the user interface software in two ways. First, future rewrites of the system which use some other DBMS package will be easier, since almost all commercially available DBMS offer SQL as a query language. Code conversion will, therefore, be relatively easy. Second, SQL is widely taught as a model database retrieval language in database design classes. Therefore, the future maintainers of the TWX system are likely to have been introduced to SQL at some point in their academic careers.

III. Review of Literature

Background of User Interfaces and Their Design

The design of user interfaces is a relatively new consideration in the field of computer science in general. Saja [17] provided an excellent summary of the history of user interfaces. He wrote that originally it was only the computer scientist himself who interacted with the computer. This is because early computer scientists were themselves the only users of their systems. This, plus the fact that the technology did not exist for interactive interfaces, made the front computer panel the only interface at that time. Later, when several scientists might need access to the computer and the various types of computer resources increased, the position of computer operator was invented. This relieved the scientists of the task of direct interaction with the computer. During this time period, interfaces were still only in the form of hardware components such as card readers. About this time, operating systems came about and presented the computer professional with somewhat more of a direct interface with the computer. Only in the last several years, as more and more untrained users have had to interact with computers, has the issue of user interface design been consciously explored. This fact, plus the continuous drop in hardware costs, has made the design of efficient, "friendly" user interfaces one of the highest concerns of the modern system builder.

For modern systems, the design of user interfaces is one of the most important aspects of overall system design. Everyone who builds systems is building a user interface, whether they recognize that fact or not. Indeed, Grimes and others have noted that "There is a challenge here to those who are actively engaged in user interface design to develop tools, methodologies and mechanisms for implementing user interface design principles early in the software development cycle. Most of all we need some successes" [8, page 25]. Some of the problems which have minimized successes in user interface design are discussed in the next section.

The Problem of Developing a Hard Science of User Interface Design

The design of user interfaces is a relatively new field which is not as developed as other aspects of computer science and computer technology. As explained before, this is because the concept of having interfaces for a computer illiterate user is itself a relatively new idea. This has caused problems for computer users that have had to tolerate systems that are not as easy to use as they could and should be. As Grimes and others have said, this is surprising in that "...over the past 10 years there has been a dramatic increase in the body of literature concerned with user interface design issues" [8, page 23]. Lack of literature is therefore not the reason that the science is not very developed.

The problem may be due to the fact that even though literature on the subject exists, the subject itself is a soft science, rather than a hard science. Newell and Card summarized the problem as it relates to computer languages as follows:

> In an appropriate science of computer languages, one would expect that half the effort would be on the computer side, understanding how to translate the languages into executable form, and half on the human side, understanding how to design languages that are easy or productive to use. Yet we do not even have an enumeration of all of the psychological functions programming languages serve for the user. Of course, there is lots of programming language design, but it comes from computer scientists. And though technical papers on languages contain many appeals to ease of use and learning, they patently contain almost no psychological evidence nor any appeal to psychological science. [14, pages 212-213]

Newell and Card believe that this is a result of Gresham's Law, which is that hard science tends to drive out soft science. This refers to the tendency of hard science considerations to overshadow soft science considerations in any discipline. The hard science here is the science of designing machines in terms of architecture, electronics, memory, and related disciplines. The soft science is the science of designing human-to-computer interfaces. Part of the reason that the design of human-to-computer interfaces is considered soft is that consideration of human factors requires a relatively subjective approach to a design problem. This subjective approach can often be a difficult task [14].

A first step in hardening the science of user interface design is the prerequisite hardening of the science of user psychology. In Allen Newell's address to the Computer-Human Interaction Conference in 1985, he presented a description of his vision of a hard science of user psychology. His vision had the following elements, as described by Thomas Moran in the July 1985 edition of the Special Interest Group in Computer-Human Interaction (SIGCHI) bulletin:

> It focuses on design, not evaluation. It has the form of an engineering-style theory based on task analysis, calculation, and approximation. It is used by interface designers at design time. [13, page 8]

In Newell's address, he said that the approach of actually designing a hard science of user psychology has several problems; among them that the science is too low-level and too slow to keep up with the more rapid advances in design. His message was that the task of hardening this science would not be easy, but that some models had already emerged. He asserted that these models need to be filled out and added to. He felt that this approach of hardening the science of user psychology was the best way to contribute "...to a science base that raises the quality of the design of human-computer interfaces" [13, page 8].

In spite of the softness of the science of user psychology, there are models for developing interfaces. These models are based on a study of the psychology of the interaction between humans and computers.

Models for User Interface Design

One of the most basic modeling approaches to user interface design is the use of the cognitive model. A cognitive model is simply the user's internal, mental representation of the interface. Note that this model will be based on the interface only, not on any of the machine processing occurring outside of the interface.

During the development of a system, the system builder develops his own model for the way he wants the system to appear to the user. This model is referred to as the builder's conceptual model. The builder must also consider the user's cognitive model during the development of user interfaces. Saja has noted that the interface comes between the builder's conceptual model and the user's cognitive model. He explained:

> In developing a system, particularly the software interface, the software engineer (who will be referred to as the designer) forms a conceptual model of the system. This conceptual model provides the designer with an understanding of the system functions and the constraints on these functions. The development of the conceptual model is an accepted phase of the software development process. Another, much later phase, occurs when a person interacts with the system. [17, page 36]

One accepted way for the designer to help the user build a workable cognitive model of a system is based on the idea of a consistent system "myth." Rubinstein, Hersh, and Ledgard defined this idea when they wrote:

> To ensure ease of learning, a system must have consistent external behavior - behavior independent of its internal workings. We like to refer to this concept as the presentation of a consistent external myth. We call it a myth to emphasize that what the users actually see may not relate directly to the internals of the system - bytes, files, control sections, jobs, links, and so on. [16, page 9]

This myth helps the user feel as if he is in contact with something that he already understands. Hutchins, Hollan, and Norman call this "directness." They state " Directness is an impression or a feeling about an interface" [10, page 317]. These authors feel that this feeling of directness "...results from the commitment of fewer cognitive resources" [10, page 317].

For the TWX user interfaces, the myth presented to the user will simply be that he is not working with a computer or DBMS, but is rather only doing the work of mission and logistics planning. The best way to promote this myth is to make the interface itself as transparent to the user as possible. Ideally, using the interface will be even easier to use than if the mission and logistics planning were being done only on paper.

Note that both the cognitive and myth models stress human factors in interface design. Making at least a cursory review of human factors considerations is essential to developing a sound interface.

Human Factors in User Interface Design

Taking human factors into consideration in user interface design means building the interface based on psychological and physiological characteristics of the potential user. A complete study of human factors as they relate to any particular field involves the disciplines of ergonomics, psychology, and task analysis.

When the issue of human factors in user interface design is discussed, a natural first question is whether or not systems designers take human factors into account in the development of modern systems. A survey by Grimes and others showed that there was "...a long way to go before human factors issues are universally well integrated into the design process" [8, page 25]. They further go on to note that this is surprising given the fact that for the private sector, over 50 percent of the cost of software development is devoted to the user interface [8].

One of the reasons that human factors are hard to integrate into the design process is that human factors considerations are the softest part of the science of user interface design. As discussed earlier, soft science considerations tend to be driven out by hard science considerations in any field. This does not mean that human factors are not important in user interface design. They are, in fact, becoming ever more important as the number of inexperienced systems users grows [17]. A complete analysis of all potential human factors in the design of the TWX interfaces is beyond the scope of this thesis effort. However, a review of some of the more general principles involved in designing for human factors will aid in the development of a sound user interface.

The characteristics that make a user interface comfortable to use are a very important aspect of designing for human factors. Comfort here refers to both the physical and psychological comfort associated with using an interface. The ACM Special Interest Group for Software Engineering (SIGSOFT) has proposed ten desirable characteristics for user interfaces [4, page 106]. These characteristics will help make interfaces comfortable to use. This group believes that a user interface should be:

1. open-ended

2. multi-modal

3. easy to learn and use

4. programmable

5. ergonomically comfortable

6. forgiving

7. consistent/uniform

8. customizable

9. able to offer immediate help

10. good in its response-time characteristics

Each of the above terms will be developed further and related to the TWX user interface design.

Open-ended refers to the ability of an interface to absorb new capabilities and functions added to the system. A user interface should be written so that upgrades to the system will appear consistent and not seen as afterthoughts to the user. One way that this concept is being applied to the TWX user interface is through a "hook book." This hook book will be a part of the program maintenance manual for the user interfaces. The hook book merely states where certain required upgrades would most appropriately fit into the interface. If any of these requirements are known, but have not been implemented for some reason, they will be recorded in the hook book. An example is the capability to actually play the land battle portion of TWX. Currently, the land battle is started with certain orders of battle for both Red and Blue. No user control is allowed over the land units once the game is started. A highly desirable part of the system here would be a user interface to the land battle portion of the exercise. This fact will be noted in the hook book. Planning for upgrades in TWX is important, as will be explained in more detail later.

Multi-modal refers to the ability of an interface to adapt to the level of expertise of the user. Menu-driven systems have traditionally irritated users who were experts on the system and who wished for a more flexible interface. The TWX user interface will not be multi-modal for two reasons. The first is that time constraints make it impossible to develop more than one user interface at this time. The second is that for the Blue TWX player, a faster interface would never be needed. This is because the blue players only have five sessions with TWX, hardly enough to become experts on the system. However, a multi-modal capability for the Red player interface would help the Red players tremendously. This is because the AU personnel who are full-time Red players definitely pick up the expertise necessary to use a faster interface. This is an area for further research and system development in the future.

Easy to learn and use are self-explanatory terms. These characteristics are those which the builder strives the hardest to build into a system during user interface design. The new TWX user interface will hopefully be considered easy to learn and use. One fact concerning TWX play makes this a probability. This is the fact that TWX players are to have their input more or less ready before the computer session is begun. Therefore, although editing capabilities are available for many parts of the interface, little editing should be required. Because of this prior planning by the user, TWX sessions will tend to follow a standard pattern, making the interface easy to learn.

Being programmable refers to an interface being adjustable by the end user. This refers to such things as users defining their own macro commands and wildcards. Due to time constraints, the TWX user interface will not be made programmable. Once again, the lack of this capability will not affect the Blue team for reasons mentioned earlier. However, this capability would be very useful to the Red players, also for reasons mentioned earlier. This is another area for further system development in the future.

Being ergonomically comfortable refers to the physical comfort associated with an interface. The TWX user interface is tied to the Z-158 hardware in this regard, and this is an aspect of the user interface that cannot be altered at this time. However, it appears from experience that using the Z- 158 does not cause any back, finger, or eye strain. This will probably be the case for the TWX users as well. The main factor in determining whether a particular terminal session would be comfortable or not would be the length of time for the session. Obviously, interfacing with a computer terminal for several hours at a time will be uncomfortable to some degree.

Forgiving refers to an interface's ability to handle user errors. This ability of a system can be measured in degrees. The least that a system should do is not to abort or halt the terminal session after a user error. A higher level of interface forgiveness would be when the system gives a user multiple chances to make a correct input. A still higher level of forgiveness is that associated with a pure selection style system, where the user has no options other than correct ones. The highest level of forgiveness occurs with user modifiable systems. A modifiable system is one in which the system allows the user to correct system errors and create input options on his own. This obviously presupposes some expertise on the part of the user. However, this style of interaction is clearly the most forgiving possible. The TWX user interface will be forgiving in that errors made in entry, either related to syntax or integrity, will immediately be flagged by the system and brought to the attention of the user.

Consistency and uniformity are the characteristics of an interface that help a user to easily conceptualize the present state of the system. These characteristics relate closely back to the concept of the user's cognitive model. Here, consistency and uniformity mean that an interface will have the same functions defined for the same terminal keys across all of the interface screens. For example, the key to get help should remain the same across the entire interface. The TWX user interface will have consistent features of this nature.

Being customizable refers to an interface's ability to be adapted to individual users. This concept is closely related to the earlier concept of programmability. Here, however, the factors that are being altered are such things as speaker volume, blink rates, and font size. Here again, the

TWX interface will be limited in this regard to customizability functions provided by the Z-158 computers themselves.

Immediate help is a self-explanatory term referring to the ability to access help features from any point in the user interface. Ideally, this would mean a help screen being available during any particular screen of the user interface. The TWX user interface will have this feature. Almost every screen in the interface will have a help screen which will explain permissible input as well as the flow of control associated with the screen. This is in addition to several memory aid screens in the form of windows which will come up automatically during certain times in the TWX computer session. These memory aid screens will contain data that will help the user to make valid input. Since the user will have largely determined his input before beginning the terminal session, these memory aids will serve as confirmation of the validity of the user's input. In places where the user would have unintentionally made invalid input, these memory aids will alert him of this possibility before the input is actually made. An example here is that during mission planning, a window will appear informing the user of how many sorties he has available by aircraft type. If the user makes bad calculations of sorties before beginning the terminal session, he will be able to adjust his plans based on the information contained in this window.

The final characteristic mentioned by the working group is that the interface should have good response time. Again, this factor for the TWX system is closely tied with the capabilities of the Z-158. Partitioning the system to keep much of the interface in memory, if possible, will help improve response time. However, not all of the database information used by the interface will fit into main memory at once. In this case, the limiting factor will be the speed of the interaction with either the Microvax or the cartridge style Bernoulli disk drive.

The focus of much of the work in this area is on general guidelines and principles for interface design. Another excellent source here is Woffinden, who did a survey of literature concerning design principles for user interfaces [19, page 20] . From these, he culled a list of "General Design Principles," which were:

1. Determine the purpose of the system
2. Know the user
3. Identify resources available
4. Consider human factors
5. Determine the interface language
6. Consider the environment of operation
7. Design for evolution
8. Optimize training
9. Accommodate levels of experience
10. Use selection vs. entry
11. Be consistent
12. Anticipate errors

Woffinden also broke down the idea of an interface into two parts, the physical interface and the psychological interface. The physical interface refers to the physical characteristics of users, such as manual dexterity and body dimension; as well as the physical capabilities associated with computer hardware. The psychological interface is that part of the interface dealing with the psychology of the user [19].

Woffinden's principles are an approach to the issue of design. Another set of principles is from Rubinstein, Hersh, and Ledgard [16, pages 219-221] . They introduced a set of 93 guidelines to follow when going through the design process. Their approach was to have a set of principles to be applied to whatever design methodology was used. They summarized their principles and guidelines in ten basic ideas:

1. Know thy user. The user is always right.
2. Develop a use model for the system.
3. Choose a good external myth to influence the way in which users think about the system.
4. Use human conversation as a model for dialogues.
5. Do documentation as part of design.
6. Build consistent human interfaces.
7. Design the errors in a system.
8. Respond to people in real time.
9. Use the user's representation for data.
10. You build it, you test it.

One aspect of any design methodology is the translation of requirements to a software design specification. The representation of the structure, flow, and control aspects of the software at this stage may take any of several forms. Yau and Tsai surveyed the various representative techniques currently used. They categorized the detailed design representations as either graphical representations or language representations [20]. For the TWX user interfaces, a language representation would seem to suffice. This is because the interfaces do not have very much associated processing other than database updates triggered by user input.

An important part of any design methodology is the verification and validation of the system. Yau and Tsai have suggested that there are seven software properties that we wish to verify and validate [20, pages 717-718]. They are:

1. Completeness: A design is complete if each part of requirements specification is fully developed.

2. Consistency: A design is consistent if no conflict exists among portions of the design.

3. Correctness: A design is correct if its input and output relation can be proved true or false.

4. Traceability: A design is traceable if terms in a design have antecedents in earlier specifications.

5. Feasibility: A design is feasible if it can be implemented and maintained so that the expected life-cycle benefits of the system would exceed the expected life-cycle cost of the system.

6. Equivalence: Two designs are equivalent if they have the same behavior.

7. Termination: A design is terminated if it is sufficiently detailed for implementation.

Most methodologies contain an element of testing. Verification and validation criteria should be established and documented prior to testing. These criteria are based largely on standards established for the system during the requirements gathering and analysis stage of the life-cycle. Verification, validation and testing considerations which are specific to the TWX effort will be discussed in Chapter 6.

The following section reviews several of the prominent software design methodologies. From these, a design methodology for the TWX user interfaces will be developed in Chapter 4. Human factors considerations will be integrated into each of the various aspects of TWX user interface design which are to follow.

Design Methodologies

The design of the TWX user interface is a software engineering problem. As such, the initial review of design methodologies must be in the area of software engineering paradigms. Three of the paradigms which relate to the design of the TWX user interface are the classic life cycle, prototyping, and the use of fourth generation language (4GL) techniques. Additionally, a methodology from the area of designing decision support systems (DSS) is of special interest.

The Classic Life Cycle. The classic life cycle is the traditional system development paradigm [15, page 20]. The development of software through the classic life cycle consists of the following steps :

1. System Engineering and Analysis
2. Software Requirements Analysis
3. Design
4. Code
5. Testing
6. Maintenance

The above steps are self-explanatory. The first two deal with establishing what the system should do, and then defining the role of software in the system. Design, code, and testing are where the software is actually developed and modified to satisfy the previously established requirements. The maintenance phase encompasses all activities that occur after the system is delivered.

The classic life cycle has come under attack for several reasons. One of the main reason is that dialogue with the end user is cut off after the second step. Experience has shown that rarely does the user know all of his requirements in advance. This means that many desired functions are never put into the system, or that they are added during the maintenance phase, which is very expensive. Because user satisfaction is a very important aspect of the TWX user interface, the classic life cycle will not be used for this project.

Prototyping. A second paradigm which is suggested by Pressman is that of prototyping [15, page 23]. Prototyping is especially useful for systems in which there is to be an active and ongoing user interface. The steps in prototyping are:

1. Requirements gathering
2. Quick design
3. Build a prototype
4. Evaluate and refine requirements
5. Engineer product

The main aspect of this paradigm is that a simple, initial system is developed and tested by the user. This initial system may just be user interface screens with no supporting processing taking place. The user then evaluates the prototype. This evaluation often leads to the explicit stating of additional requirements and the refinement of existing requirements. This may occur several times. After a stable set of requirements has eventually been established, the software is engineered.

This methodology has many attractive features as far as designing the TWX user interfaces is concerned. Since the interfaces must be screen oriented, the use of sample screens to refine requirements would be a logical part of the total design method. Aspects of prototyping will be incorporated into the TWX user interface design methodology.

Fourth Generation Language Paradigm. A third design paradigm presented by Pressman is associated with the use of fourth generation languages [15, page 24] . These languages allow the software developer to specify software characteristics at a higher level than with more traditional programming languages. The steps in this paradigm are:

1. Requirements gathering

2. Design strategy

3. Implementation using fourth generation language

4. Product

As discussed earlier, one of the reasons for choosing the INGRES DBMS was that an applications generator tool would be available. This tool is essentially a fourth generation language. For this reason, the TWX user interface design methodology will incorporate the aspect of fourth generation language implementation from the above paradigm.

Iterative Design. Another methodology of interest here is that of iterative design. This design method was described by Sprague and Carlson in their book "Building Effective Decision Support Systems" [18, page 140]. A Decision Support System (DSS) is a computer based system which is used to assist the user in making unstructured decisions. What is of real interest here, though, is the fact that DSS must have usable, effective, user interfaces. This is because of the nature of the tasks that DSS must help the user to do. Because of this fact, Sprague and Carlson's methodology of iterative design is especially applicable to systems which are user interface "intensive." The steps in this methodology are:

1. Identify an important sub-problem.

2. Develop a small but usable system to assist the decision maker.

3. Refine, expand, and modify the system in cycles.

4. Evaluate the system constantly.

Note that the emphasis here is on the addition of new capabilities to the system over time. This is because the DSS must be flexible enough to adjust to the user's decision making style. This aspect of expanding over time is also applicable to the TWX user interfaces. Many desirable functions will not be implemented with this effort simply due to time constraints. Those functions that are already anticipated to be added in the future will be noted and written down. These functions may then be added by personnel at the Wargaming Center as needed.

One of the objectives of all of the various life cycles and methodologies is to develop modular software. Ideally, user interface software should be a self-contained module. This modularity is beneficial in that it will allow the interface to be modified based on user evaluation without affecting other areas of the system. Another benefit of stressing modular software for the user interface is that it allows the system builder to be consistent throughout his design [17]. For these reasons, the TWX user interface will be developed in a modular fashion, separate from the other software used in the system.

Having reviewed various methodologies for the design of software, methods for making the transition to actual code must be considered. The following section deals with the implementation to software of a system design.

Theoretical Aspects of Implementation

Once the actual conceptual design model has been created, the implementation to software must be done in such a way that insures that the resultant system is actually what the designer intends. Mark described a methodology which may be used to help translate the given hardware and software capabilities that the designer has at his disposal into the interface design that is desired. He said that "The crucial first step in design methodology is to get programmers to state their design models in terms of a set of ideas that the system already understands" [12, page 341]. Only from this explicit mapping of a conceptual design model into existing system capabilities is the design of a correct interface possible.

For screen-oriented user interfaces such as TWX, sample screen designs serve as the starting point for actual system implementation. In cases where screens cannot be implemented as originally planned due to system constraints, adjustments are then made as necessary. The actual format of the screen designs is especially important at the implementation stage. Format of screens is addressed by Galitz in his book "Handbook of Screen Format Design" [7]. The advice given in this book may be applied during both the requirements analysis stage (to the sample screens), and during the implementation stage of the project.

Among the aspects of screen design that Galitz gives guidelines for are: wording, headings, spacing, font size, tabbing, graphics, and color. The guidelines given throughout his book are designed to help the system builder insure consistency and ease of use in screen design [7].

An important aspect of implementing user interfaces is giving the user status and feedback as part of the screens. Saja has noted that this information allows the user to keep track of his position in the system, and builds the user's confidence as well [17, page 38]. Messages to the user will also inform him when his input has been correct and the system is processing.

Another important aspect of implementation is the selection of appropriate output representations to aid the user. This topic is reviewed in the next section.

Output Representations

Related to the previous ideas is the question of which format of computer output is best in terms of helping the user understand the processing that the system has done. The user should be able to make decisions regarding his application without regard to its specific implementation. In environments such as that of TWX environment, users have to make decisions based on computer generated output. The output must be easily understood and applicable to the situation. Dickson, Benbasat, and others obtained experimental data concerning the issue of whether or not computer graphics provided any greater benefit in terms of such an understanding than more standard tabular output [2]. Their experiments were carried out in the hopes of finding out the type of output which assisted most in the discovery of an optimal solution to a hypothetical problem. These sources generally agree in their conclusions.

Benbasat, Dexter, and Todd wrote "To summarize, graphical presentations are expected to assist at the earlier stages of the problem-solving process by pointing to the general region and direction of the more profitable outcomes. However, their value and use will be reduced when the decision maker is making marginal adjustments to reach the optimal solution" [2, page 77].

Dickson, DeSanctis, and McBride expressed the same conclusions when they wrote "In tasks for which accurate interpretation of values is important, especially in environments in which the user has some experience with tabular presentations (such as dealing with financial data), tables are probably a better choice of format than graphs" [6, page 46].

Since the TWX output will be used to make decisions regarding the next day's logistics and mission planning, tabular format is used for the reasons above. Note in this case that officers playing TWX are usually from an organizational level in the Air Force where they are most likely very familiar with tabular and textual status reports.

The Future of User Interfaces

Having reviewed some of the theory on user interfaces, the design principles used in their development, and other associated aspects, the question arises as to where the future of user interfaces is headed. One of the areas of particular interest is the question of whether the user interface of the future will even be screen-oriented at all. Recall the cognitive model that users build for a system, discussed earlier. There are design cases where the screen may not be the best way to present objects to the user. Bolt stated that the terminal screen used for present interfaces is insufficient in many ways. His argument relates closely back to the concept of the user's cognitive model for a system. Bolt wrote:

> At the typical terminal, all transactions - invoices, mail, personal notes, whatever - occur in the same spot. A succession of things appear, each new item literally displacing the old. Further, on the majority of current terminals most messages look alike: lines of print. There is nothing distinctive about a message as it is embodied or presented that helps recall. There are just not enough distinguishing circumstances surrounding the message to make it memorable on any dimension except content. And if the message's

> content is not vivid or meaningful, then it tends to lose itself in the general stream of sameness. [3, page 82]

In other words, much of what we do, think about, and remember in everyday situations "...is 'tagged' in memory with the time it occurred plus incidental particulars" [3, page 82]. The terminal, then, does not appear to be a device which aids in this tagging process. Therefore, the user interfaces of the future may try to take advantage of input and output devices other than the terminal. These devices will present data in a way that aids in remembering the context of the data itself.

IV. Design and Implementation of the TWX User Interfaces

The TWX Design Methodology

After considering the many possibilities for design metho^dologies previously described, a methodology for developing the TWX user interfaces was developed. This methodology combined the main features of prototyping, fourth generation techniques, and iterative design. The steps in this methodology are:

1. Identify initial requirements through review of the current system, sample screen designs, and interviews with Wargaming Center personnel. Identify desired functions which will not be initially implemented in a hook book.
2. Make an initial prototype using the fourth generation language, with only a small supporting database, and no simulation processing.
3. Let Wargaming Center personnel evaluate the prototype.
4. Refine the prototype based on user suggestions.
5. Create an initial system by combining the prototype with the new TWX database system.
6. Combine the system from step 5 with the air and land battle simulations.
7. Add new functions and evaluate the system iteratively.

Due to time constraints, step 7 responsibilities will fall onto the Wargaming Center. The hook book will aid in deciding where in the software to add the new functions.

One of the most important part of the methodology was the design of sample screens. These served as the documentation for the requirements analysis and gathering phase. These sample screens contained both a pictorial representation of the screen itself as well as text describing the screen. This text had six basic sections: control flow, input fields, integrity constraints, processing, screen locking, and function. Each of these sections will now be described further.

Control flow refers to how a user may enter and exit from the screen. It includes the names of the screens which the user may enter the screen from, and the screens which the user may exit the current screen to. Note that this includes all menus and help screens associated with the screen as well.

Input fields refers to the possible user input for the screen. This text will describe all fields which the user may input and the results of each input. Certain TWX screens require selecting from alternatives rather than inputting values. Therefore, descriptions of the various selection alternatives are included here as well.

Integrity constraints refers to the allowable values for each of the input fields and selection alternatives. Documenting these integrity constraints assists in developing the help screens for the user interface, as well as assuring proper input values are input through th: user interface software.

Processing refers to the processing that must occur in the software as a result of each input. For a major part of the TWX user interface, input processing only consists of updating relations in the database with the user input. Areas in which this is not the only processing which must occur will be especially noted.

Screen locking refers to integrity checking at a higher level than the field level. Two users cannot be allowed to attempt to update the same relation in the database at the same time. The screen locking text for each screen will therefore describe the screens that other users in the same seminar may access while this screen is being used by a particular player.

Function refers to the overall purpose of the screen. This will include a description of the game functions that the screen gives the user access to, such as mission planning or logistics movement.

This documentation method was chosen for the requirements analysis of the TWX user interfaces for two reasons. The first is that this documentation allows for a quick transition to a prototype by using the fourth generation applications generator. The second reason is that this method provides adequate documentation of the processing that occurs, since little processing outside of database updates is required for the user interfaces. The fact that not much processing needs to be documented is why this method was chosen over other methods which stress the flow of processing, such as data flow diagrams or flowcharts.

The method of gathering user requirements and other requirements considerations have been discussed in various sections of this paper to this point. The next section explicitly summarizes this stage of the TWX user interface development.

Requirements Analysis Process

Requirements analysis included finding and analyzing user requirements for both information and control flow. As far as the information requirements were concerned, they were largely the same as those of the current system. This is because the actual play of the game and the decision making processes involved needed to remain the same. Since the use of hard copy devices for input was no longer a constraint, many of the input formats were modified in order to allow the exercise players to more efficiently make their logistics and mission planning decisions. Analysis of control flow involved finding out the best sequence of screens to allow the user to input data easily. For instance, editing features which allow for changing the input on a previous screen were highly desirable for TWX players. In-person discussions with Air Force Wargaming Center personnel were essential for the completion of the requirements analysis phase.

The actual inner logic of the exercise was not changed. Currently, TWX's air and land battle simulation programs receive data based on user input. These programs then perform complex calculations which simulate the theater conflict. These calculations determine results from input decisions in the areas of logistics, doctrine, and warfighting principles. These programs then output results of the battle to the user. The formulas and calculations which determine the battle results must remain the same even as the programs of which they are a part are rehosted from a mainframe to a microcomputer.

The screen design phase involved drawing each user input and output screen. These were first done by hand and then generated by the INGRES Visual Forms Editor program. This program allowed the generation of an initial prototype without having to connect the user interface screens to any network or simulation software.

Together with added control flow, the screens designed constituted a first prototype which was evaluated by Wargaming Center personnel. Their suggestions were incorporated into a more final design which was then implemented with the new database system.

Developing the TWX user's guide is the responsibility of the Wargaming Center, and will follow the full implementation of this project. The user interface software maintenance manual also will be written only after the code itself is in a reasonably final form.

A Review of the TWX User Interfaces

A few definitions are necessary in order to fully understand the play of TWX and the user of the user interfaces which follows. The entire discussion which is to follow deals with the Blue side only. This will be sufficient to illustrate what is done in a typical TWX computer session. TWX seminar groups actually play two separate roles: those at the AAFCE (Allied Air Forces Central Europe) level, and those at the ATAF (Allied Tactical Air Force) level. The ATAF level is divided into the 2ATAF and 4ATAF. Team members have a role in each of the two levels. The "Theater Warfare Exercise Handbook" [1, page 6] provides a summary of the relationship between the two levels and the direction of TWX play:

> Players should also note ... that Blue (and Red, although Blue terminology will be used) will be making decisions at two different levels. The first is that of the Commander, Allied Air Forces Central Europe (COMAAFCE), who in concert with his staff, develops and air strategy that will support the strategy of the theater commander, the Commander in Chief, Central Europe (CINCENT), represented by the control team. COMAAFCE implements his strategy by issuing an Air Directive (AD), which provides guidance to lower command levels.
>
> The next level of decision making is that of the Commander and staff of the Second and Fourth Allied Tactical Air Forces (COMTWOATAF and COMFOURATAF). At

this level, players implement the AD and make decisions to insure the optimum use of their limited assets in meeting COMAAFCE's priorities and specific objectives.

For the TWX players, this essentially means that aircraft movement, aircraft rerole, and logistics movement (the general term used is "posturing of forces") is accomplished at the AAFCE level. Mission planning, packaging, and targeting are done at the 2ATAF and 4ATAF level. The mission planning, of course, is supposed to be in direct support of the AD. Mission planning is done for each of two daily "cycles", the day cycle and night cycle. These merely represent when the planned mission is to fly.

The walkthrough, discussion and review which follows is not of every screen but a representative sample of all screen types. All of the processing associated with each screen will obviously not be covered here. The computer session input may not occur in the order presented. Usually, the aircraft movement and aircraft rerole portions of the AAFCE level are accomplished first. This is because of locking considerations which will be discussed later in Chapter 5. The remainder of the AAFCE input and the 2ATAF and 4ATAF input then follow. Remember that several players may be making input for the same seminar group at a time.

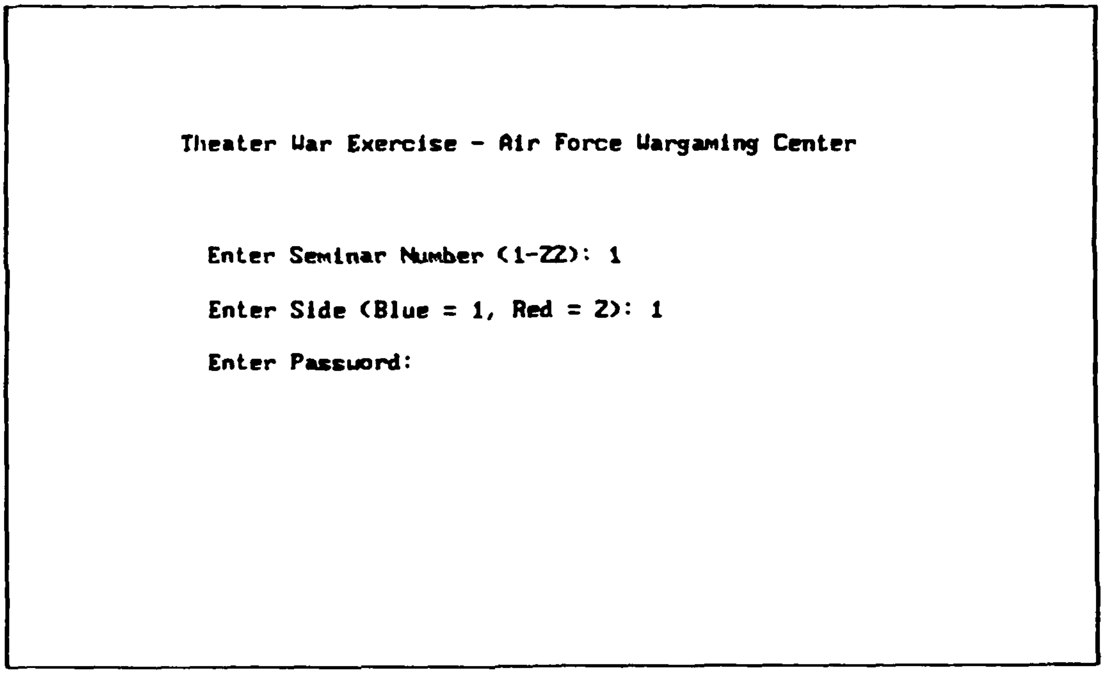
The TWX Password Screen. A password screen is always the first screen that is presented to the user during a TWX computer session. The TWX Password Screen is shown in Figure 3. There are two reasons why a password screen is necessary. The first reason is that the control team must be able to lock users out of the interface after a certain period of time. Experience at the Wargaming Center has shown that some users would work for far too long a period of time in an effort to optimize their input. To enforce reasonable time constraints, the controller team may lock out any user from logging onto the system after a certain period of time. The second reason is that only the specific seminar group should be able to do their own AAFCE and ATAF input. This restriction is applied by giving each seminar group its own password.

This screen checks the user input side, seminar number and side against values stored in the database. If there is no match, the user is given another try at correct input. There is no maximum number of tries. However, the user is advised to see the seminar control group if there seems to be a problem with the password.

Introduction to the TWX Computer Session Screen. After inputting a correct password, the next screen the user sees is an Introductory Screen explaining a little about the TWX computer session. Figure 4 shows this screen. There is no user input on this screen. The user merely presses the RETURN key when he wishes to go on to the main TWX menu screen.

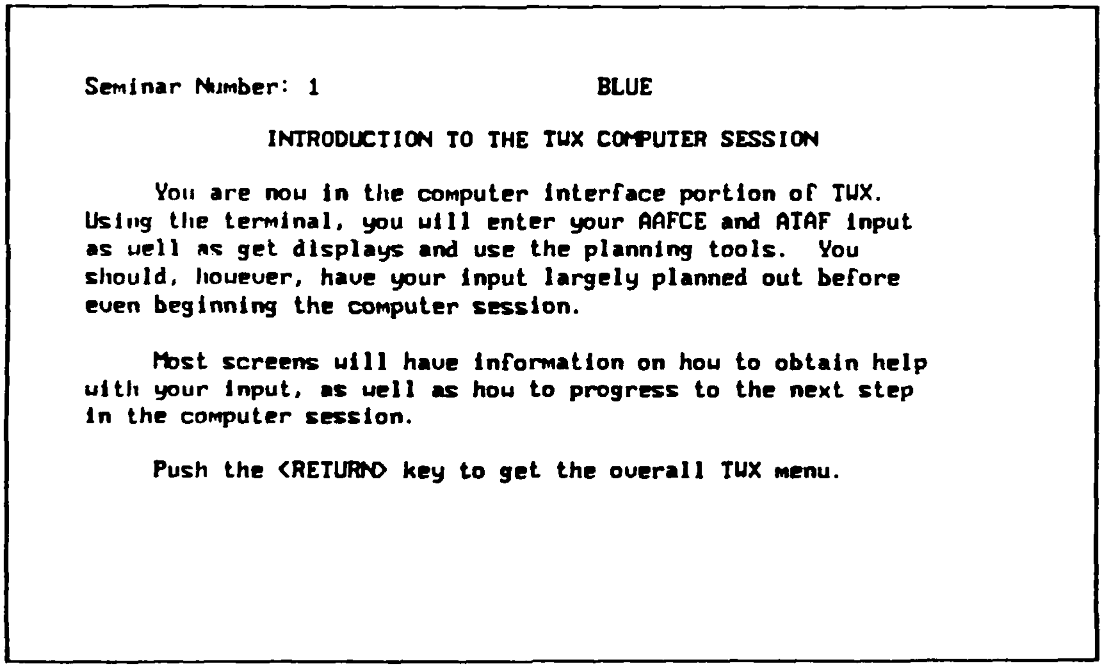
TWX Main Menu. The next screen the user is presented with is the TWX Main Menu screen, shown in Figure 5. The user chooses one of six options. Option 1 puts the user in the overall AAFCE menu. Options 2 and 4 put the user in either the 2ATAF or 4ATAF menus, respectively. Option 3 will be used to access the land interface, when one is written. Currently, option 3 only gives the user an error message and puts him back in the option field to enter another option. Option 5 is a quit key which merely logs the user of the computer. The user may come back and log back on as he wishes. This key would be used to shut down the system for a lunch break, for instance. Option 6 has a much more specific use. It is to be used only by the last person logged onto a terminal for a seminar group. This option essentially locks all of the input data, and starts the air and land battle simulations for the seminar group. Obviously, using this option incorrectly would have damaging results. Therefore, several checks are made and the user if prompted several times so that the danger of accidentally starting the simulation is minimized.

AAFCE Overall Menu. If the user selects the AAFCE option of the Overall TWX Menu, he is then placed in the AAFCE Overall Menu, shown in Figure 6. This menu has 13 options. The first eight are report options which give the user various report on the current status of aircraft and logistics items. These report are written using INGRES report writer feature. The user may have



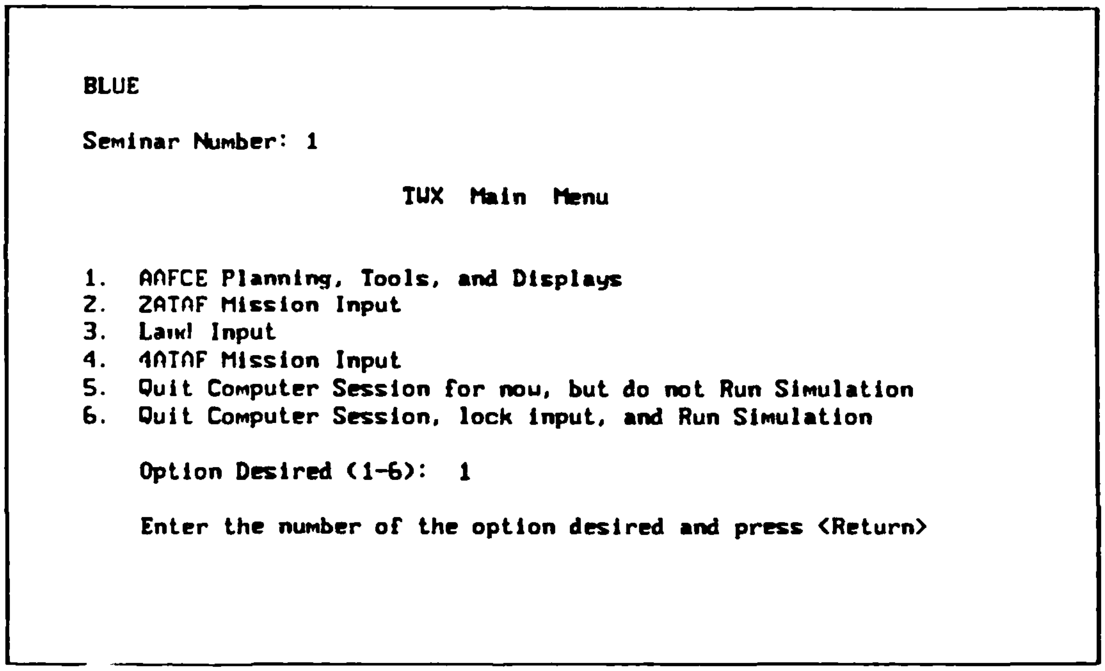
```
Theater War Exercise - Air Force Wargaming Center


  Enter Seminar Number (1-22): 1

  Enter Side (Blue = 1, Red = 2): 1

  Enter Password:
```

Figure 3. The TWX Password Screen



```
Seminar Number: 1                    BLUE

          INTRODUCTION TO THE TWX COMPUTER SESSION

     You are now in the computer interface portion of TWX.
Using the terminal, you will enter your AAFCE and ATAF input
as well as get displays and use the planning tools.  You
should, however, have your input largely planned out before
even beginning the computer session.

     Most screens will have information on how to obtain help
with your input, as well as how to progress to the next step
in the computer session.

     Push the <RETURN> key to get the overall TWX menu.
```

Figure 4. Introduction to TWX Computer Session Screen



```
BLUE

Seminar Number: 1

                              TWX  Main  Menu


1.  AAFCE Planning, Tools, and Displays
2.  2ATAF Mission Input
3.  Land Input
4.  4ATAF Mission Input
5.  Quit Computer Session for now, but do not Run Simulation
6.  Quit Computer Session, lock input, and Run Simulation

    Option Desired (1-6):  1

    Enter the number of the option desired and press <Return>
```

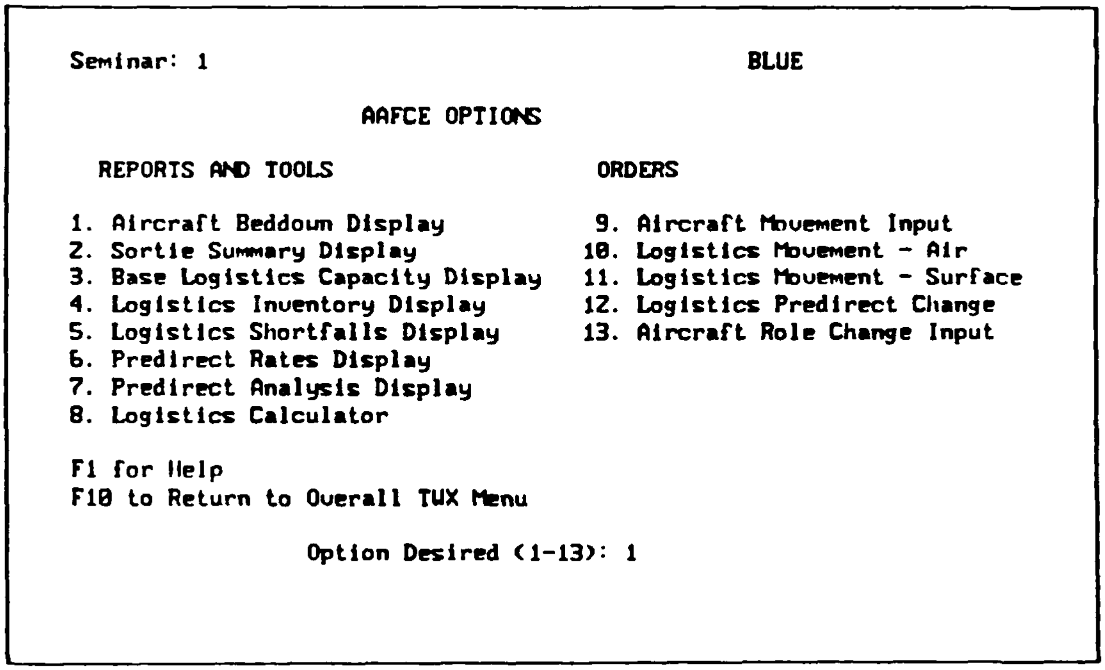
Figure 5. TWX Main Menu

several options available for each of the various reports. The reports are not true input screens and will not be discussed any further.

Each of the remaining options (9 through 13) place the user in another screen which has a specific function. Option 9 allows the user to move aircraft from one airbase to another. Options 10 and 11 allow the movement of logistics items by airlift and surface movement, respectively. Option 12 is used to change predirect rates for logistics items. Option 13 is used to rerole multi-role aircraft from one role to another. There is also a Help Screen available and Function Key F10 is used to return to the next higher menu level, the TWX Main Menu.

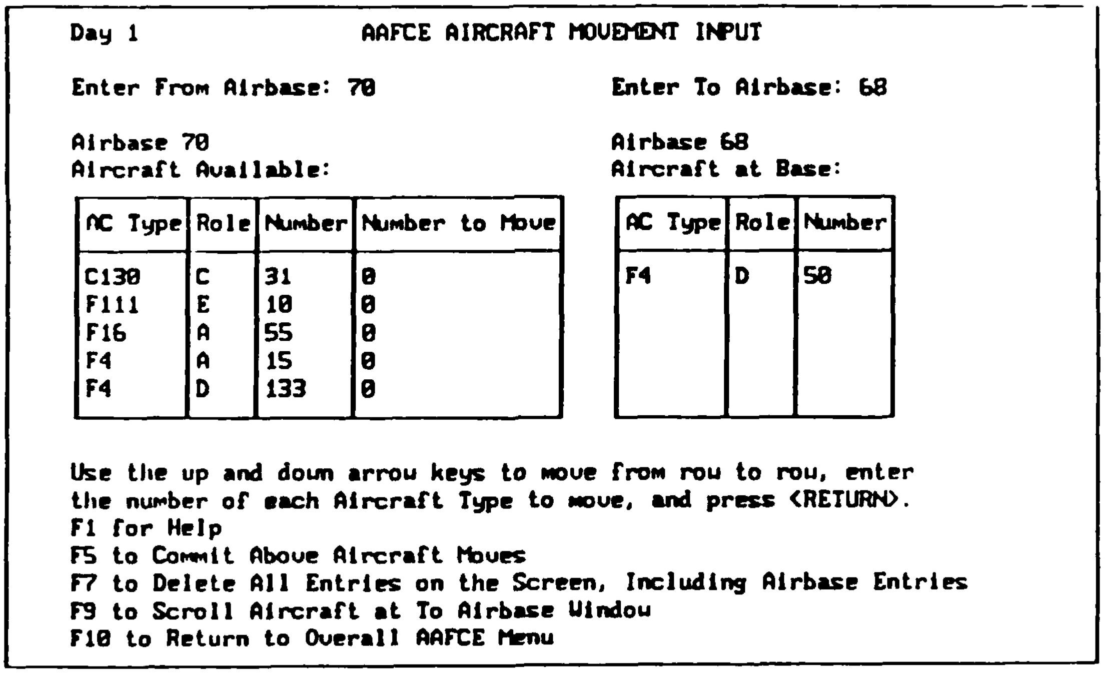
AAFCE Aircraft Movement Screen. If the user selects option 9 of the AAFCE Overall Menu, he is then presented with the AAFCE Aircraft Movement Screen, shown in Figure 7. The user first enters a "From" Airbase Number and a "To" Airbase Number. These are validated and a check is made to make sure the "To" Airbase has enough ramp space to accommodate any more aircraft. If not, the user is informed of the problem.

After entering the airbases, the user is presented with two tables of information. The first shows all of the aircraft at the "From" Airbase that are eligible to move to the "To" Airbase. An aircraft may not be eligible to be moved to another base because the receiving base may not have the maintenance support required for the specific aircraft type and role. Such aircraft will not be shown, in order to avoid confusion for the user. The user is also shown all of the aircraft at the "To" airbase, so he can get an idea of the total number and types of aircraft that would be on the base after his moves. The user then enters values in the "Number to Move" column until he is satisfied with his input. Here, obviously, values may be simply edited as desired. Only when the user pushes the F5 "Commit" key are the values actually updated in the database. As always, there is a Help Screen available, and F10 returns the user to the next highest level in the interface (here the Overall AAFCE Menu).



```
Seminar: 1                                        BLUE

                        AAFCE OPTIONS

  REPORTS AND TOOLS                    ORDERS

1. Aircraft Beddown Display           9. Aircraft Movement Input
2. Sortie Summary Display            10. Logistics Movement - Air
3. Base Logistics Capacity Display   11. Logistics Movement - Surface
4. Logistics Inventory Display       12. Logistics Predirect Change
5. Logistics Shortfalls Display      13. Aircraft Role Change Input
6. Predirect Rates Display
7. Predirect Analysis Display
8. Logistics Calculator

F1 for Help
F10 to Return to Overall TUX Menu

                 Option Desired (1-13): 1
```

Figure 6. AAFCE Overall Menu



Day 1 AAFCE AIRCRAFT MOVEMENT INPUT

Enter From Airbase: 70 Enter To Airbase: 68

Airbase 70
Aircraft Available:

AC Type	Role	Number	Number to Move
C130	C	31	0
F111	E	10	0
F16	A	55	0
F4	A	15	0
F4	D	133	0

Airbase 68
Aircraft at Base:

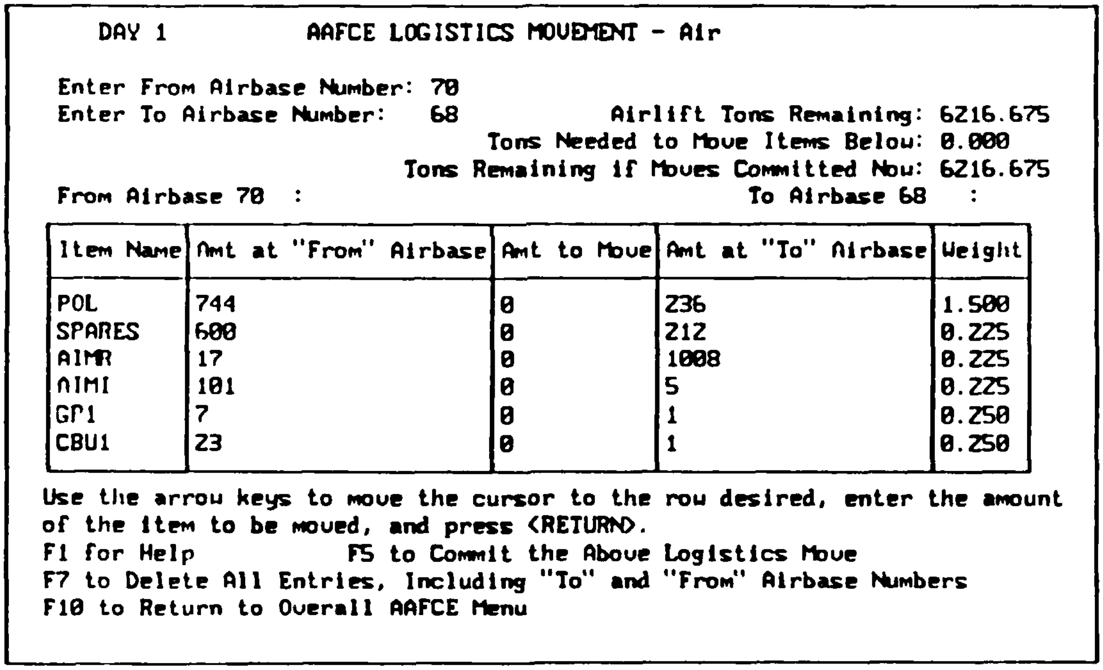
AC Type	Role	Number
F4	D	50

Use the up and down arrow keys to move from row to row, enter the number of each Aircraft Type to move, and press <RETURN>.
F1 for Help
F5 to Commit Above Aircraft Moves
F7 to Delete All Entries on the Screen, Including Airbase Entries
F9 to Scroll Aircraft at To Airbase Window
F10 to Return to Overall AAFCE Menu

Figure 7. AAFCE Aircraft Movement Screen

AAFCE Logistics Movement by Air. The screens for AAFCE Logistics Movement, whether by airlift or surface movement, are essentially identical in appearance. They are also largely identical in function. Figure 8 shows the AAFCE Logistics Movement by Air Screen. As with the Aircraft Movement Screen, the "From" and "To" airbases are entered by the user and verified by the interface. The amounts of each of the logistics items at the two bases are shown. As before, the user only inputs values in the "Amount to Move" column. There is also a set of counters presented in the upper right hand portion of the screen. These counters give the user of status of his airlift capability,as well as how that status would change if the moves he has entered on the screen were committed. Editing can be done on the amounts in the "Amount to Move" column if desired. Once F5 is pressed, values are changed in the database to reflect the moves, including the corresponding reduction in airlift capability.

AAFCE Predirect Rate Change. The AAFCE Predirect Rate Change Screen, shown in Figure 9 is another example of the types of improvements which were made in the new user interfaces. Predirect rates are the daily airbase resupply rates for petroleum, spare parts, and munitions. In TWX, these rates require adjustment as the game progresses from day to day. First, the user enters an airbase number for the airbase he wishes to change predirect rates at. He then enters the desired new rates, committing his input when done. Although the concept is simple, the improvements over the old system are significant. With the old version of TWX, the base number and the desired predirect rates are entered over a hard copy device. No feedback or editing capability is provided to the user. For instance, if the user accidentally inputs the wrong base number before entering new predirect rates, then the input rates will be entered in the database against the specified (incorrect) airbase. With the new user interface, such a situation should not occur. A clear key exists which can be used to start the entry process over with a new airbase number. As stated earlier, no transactions are actually entered against the database until a commit



DAY 1 AAFCE LOGISTICS MOVEMENT - Air

Enter From Airbase Number: 70
Enter To Airbase Number: 68
Airlift Tons Remaining: 6216.675
Tons Needed to Move Items Below: 0.000
Tons Remaining if Moves Committed Now: 6216.675
From Airbase 70 :
To Airbase 68 :

Item Name	Amt at "From" Airbase	Amt to Move	Amt at "To" Airbase	Weight
POL	744	0	236	1.500
SPARES	600	0	212	0.225
AIMR	17	0	1008	0.225
AIMI	101	0	5	0.225
GP1	7	0	1	0.250
CBU1	23	0	1	0.250

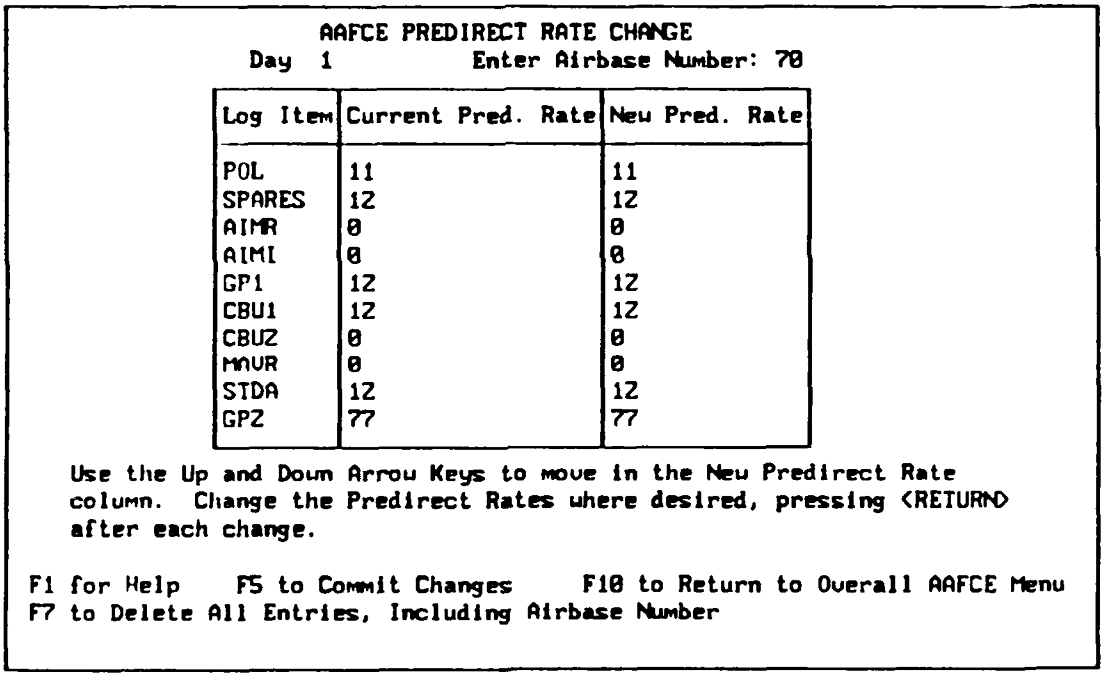
Use the arrow keys to move the cursor to the row desired, enter the amount of the item to be moved, and press <RETURN>.
F1 for Help F5 to Commit the Above Logistics Move
F7 to Delete All Entries, Including "To" and "From" Airbase Numbers
F10 to Return to Overall AAFCE Menu

Figure 8. AAFCE Log Movement by Air Screen

key is pressed. An online help screen is available. Finally, a column showing the current predirect rates for the entered base number serves as a memory aid for the user.

AAFCE Aircraft Role Change. In TWX, some aircraft are multi-role, which means they may take on another role if desired. This may be the case, for instance, if more defense aircraft are needed and there is an overage of attack aircraft. Some of the attack (A) type aircraft may be able to be reroled to a defense (D) role. The AAFCE Aircraft Role Change Screen is shown in Figure 10. Rerole, like all other aspects of AAFCE input, is done on an airbase-by-airbase basis. Once again, the first input made by the user is an airbase number. The user is then presented with a table of aircraft which are eligible for rerole. Here, it should be noted that not all aircraft are eligible for rerole. This is to reflect real-world shortfalls in maintenance capabilities. Once again, the user may only input in the "Qty to Rerole" column. The reroles are not committed until F5 is pressed. Function key F10 returns the user to the Overall AAFCE Menu. The AAFCE Aircraft Role Change Screen is the last option available from the Overall AAFCE Menu.

ATAF Overall Menu. The 2ATAF and 4ATAF Overall Menus are accessed via options 2 and 4 of the TWX Overall Menu, respectively. The 2ATAF and 4ATAF Overall Menus are identical in structure. The 2ATAF Overall Menu Screen is shown in Figure 11. This menu is similar to the Overall AAFCE Menu in that options selected here do not affect database values, but merely let the user select other screens from which missions can be input. There are nine basic mission types, which fall into three basic categories. The three categories are: missions which are targeted and may have support aircraft, missions which do not have either targets or support aircraft, and Close Air Support (CAS) missions. The first category of missions includes the Offensive Counter Air (OCA), Interdiction (IND), Battlefield Air Interdiction (BAI), and Reconnaissance (RECCE) mission types. The second category includes the Defensive Counter Air (DCA), Defense Suppression (DSUP), Combat Air Patrol (CAP), and Electronic Countermeasures (ECM) mission types. The third category includes only the CAS mission type. In each of the three categories, the screens



AAFCE PREDIRECT RATE CHANGE

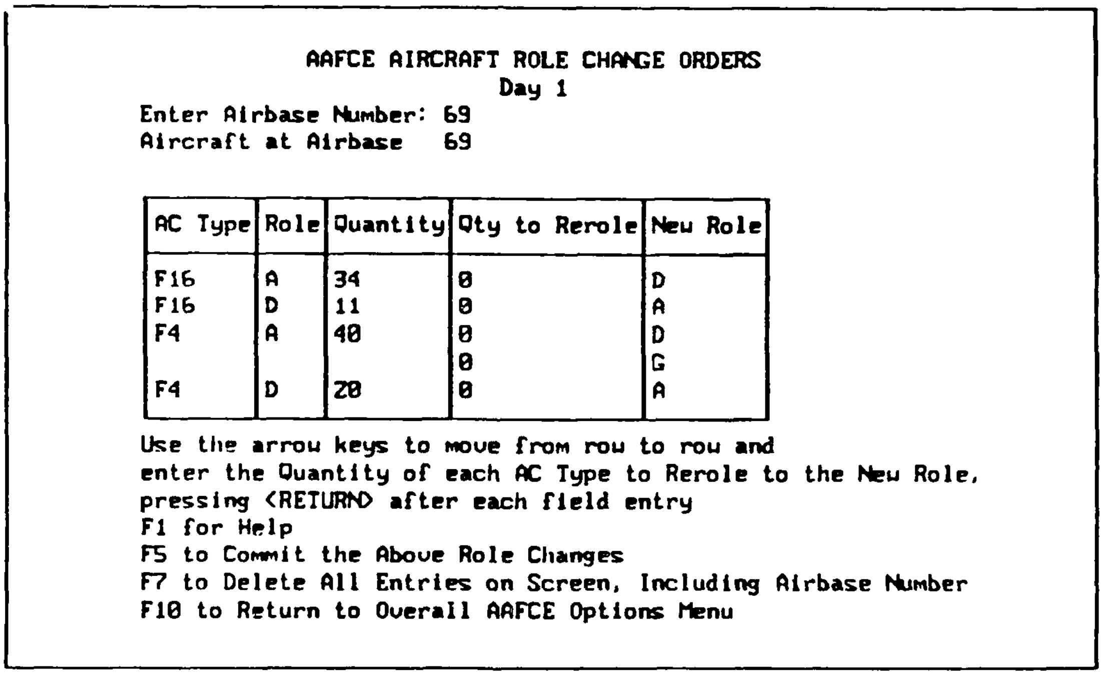
Day 1 Enter Airbase Number: 70

Log Item	Current Pred. Rate	Neu Pred. Rate
POL	11	11
SPARES	12	12
AIMR	0	0
AIMI	0	0
GP1	12	12
CBU1	12	12
CBU2	0	0
MAUR	0	0
STDA	12	12
GP2	77	77

Use the Up and Doun Arrou Keys to move in the Neu Predirect Rate column. Change the Predirect Rates uhere desired, pressing <RETURN> after each change.

F1 for Help F5 to Commit Changes F10 to Return to Ouerall AAFCE Menu
F7 to Delete All Entries, Including Airbase Number

Figure 9. AAFCE Predirect Rates Change Screen



AAFCE AIRCRAFT ROLE CHANGE ORDERS
Day 1

Enter Airbase Number: 69
Aircraft at Airbase 69

AC Type	Role	Quantity	Qty to Rerole	Neu Role
F16	A	34	0	D
F16	D	11	0	A
F4	A	40	0	D
			0	G
F4	D	20	0	A

Use the arrou keys to moue from rou to rou and
enter the Quantity of each AC Type to Rerole to the Neu Role,
pressing <RETURN> after each field entry
F1 for Help
F5 to Commit the Aboue Role Changes
F7 to Delete All Entries on Screen, Including Airbase Number
F10 to Return to Ouerall AAFCE Options Menu

Figure 10. AAFCE Aircraft Role Change Screen

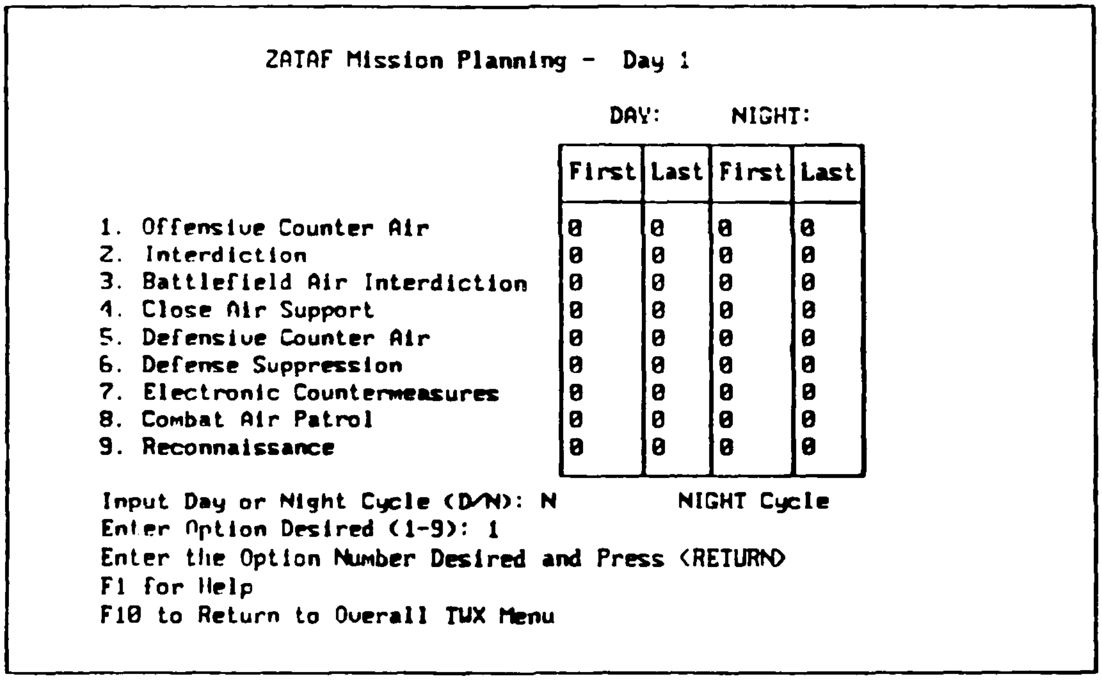
used for the various mission types are essentially identical in format. The discussion which follows is therefore just a representative sample of each of the categories of screens.

Only one person at a time may plan missions for each ATAF, to avoid database conflicts. This aspect of play is controlled by the user interface. Users input missions for both the day and night cycles. The cycle being planned for is one of the fields the user enters on the Overall ATAF Menu.

ATAF Offensive Counter Air Mission Input. The first and most complicated mission category is the targeted, supported mission category. The OCA Mission Input Screen, shown in Figure 12 is representative of the screens used for this mission category. Here, missions are assigned line numbers. This allows the user to later go back and edit mission previously input. There is a target field which the user must enter and which is verified against the database. A table shows the aircraft available to fly OCA missions for the ATAF. The user inputs the aircraft type, role and number for the mission. More than one aircraft type may be used in a single mission. Finally, the user may wish to assign support aircraft to the mission. These support aircraft are Escort aircraft, Defense Suppression (DSUP) aircraft, and Electronic Countermeasures (ECM) aircraft. This aspect of mission planning reflects real-world type mission "packaging", where support aircraft act as force multipliers to enhance the chances for mission success.

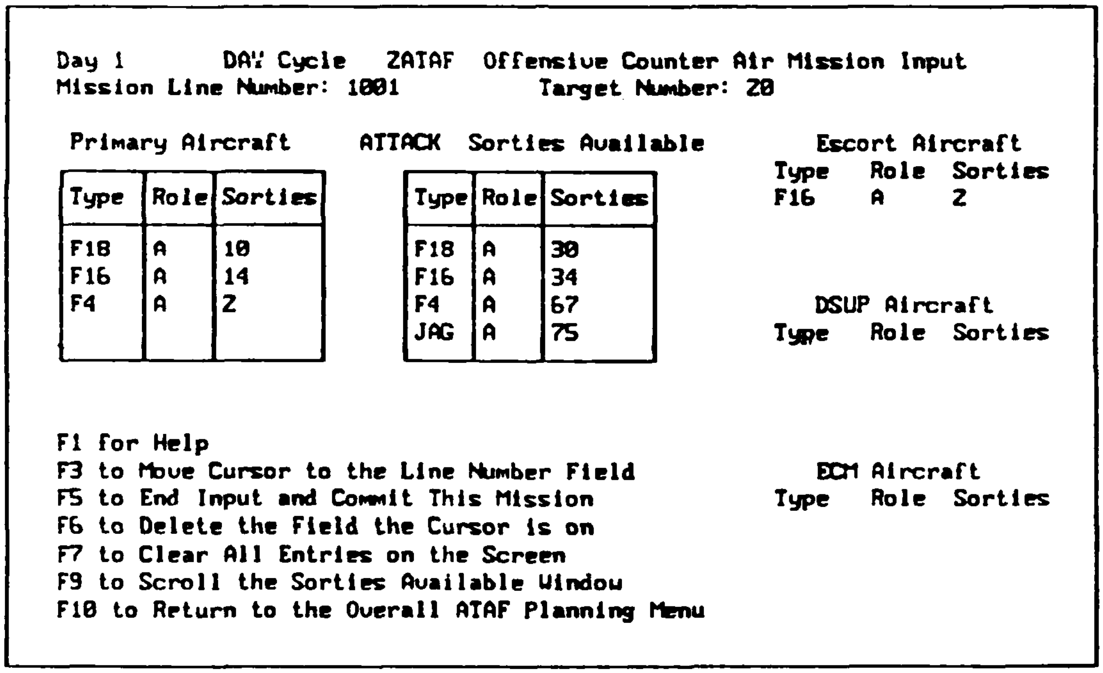
The user may edit any aspect of his mission input. When he is satisfied with the mission, he may commit it to the database. However, he may go back later and edit the mission again if he desires. When a mission is committed, all the input aircraft, including support aircraft, are checked to verify that they exist and may fly in the desired role.

ATAF Defensive Counter Air Mission Input. The second category of missions is those that are untargeted and that have no associated support aircraft. An example of the screens used with this category of mission is the ATAF Defensive Counter Air Mission Screen, shown in Figure 13.



```
                 2ATAF Mission Planning -  Day 1

                                         DAY:      NIGHT:
                                      ┌─────┬────┬─────┬────┐
                                      │First│Last│First│Last│
                                      ├─────┼────┼─────┼────┤
1. Offensive Counter Air              │0    │0   │0    │0   │
2. Interdiction                       │0    │0   │0    │0   │
3. Battlefield Air Interdiction       │0    │0   │0    │0   │
4. Close Air Support                  │0    │0   │0    │0   │
5. Defensive Counter Air              │0    │0   │0    │0   │
6. Defense Suppression                │0    │0   │0    │0   │
7. Electronic Countermeasures         │0    │0   │0    │0   │
8. Combat Air Patrol                  │0    │0   │0    │0   │
9. Reconnaissance                     │0    │0   │0    │0   │
                                      └─────┴────┴─────┴────┘
Input Day or Night Cycle (D/N): N          NIGHT Cycle
Enter Option Desired (1-9): 1
Enter the Option Number Desired and Press <RETURN>
F1 for Help
F10 to Return to Overall TWX Menu
```

Figure 11. 2ATAF Overall Menu



Day 1 DAY Cycle ZATAF Offensive Counter Air Mission Input
Mission Line Number: 1001 Target Number: 20

Primary Aircraft

Type	Role	Sorties
F18	A	10
F16	A	14
F4	A	2

ATTACK Sorties Available

Type	Role	Sorties
F18	A	30
F16	A	34
F4	A	67
JAG	A	75

Escort Aircraft

Type	Role	Sorties
F16	A	2

DSUP Aircraft

Type	Role	Sorties

ECM Aircraft

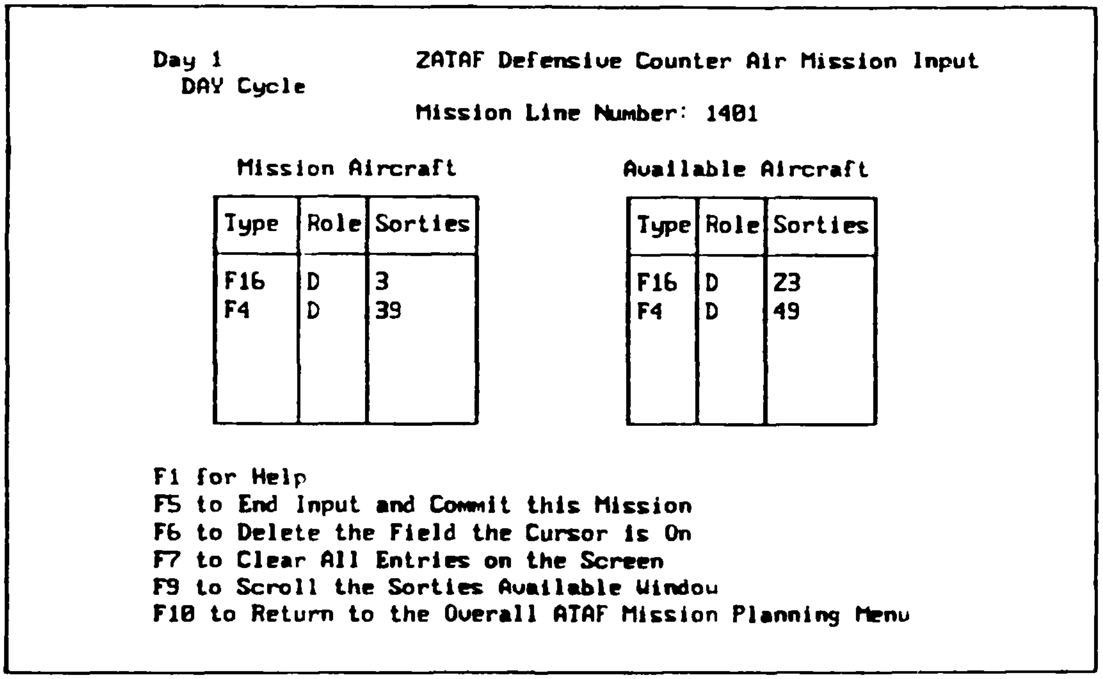
Type	Role	Sorties

F1 for Help
F3 to Move Cursor to the Line Number Field
F5 to End Input and Commit This Mission
F6 to Delete the Field the Cursor is on
F7 to Clear All Entries on the Screen
F9 to Scroll the Sorties Available Window
F10 to Return to the Overall ATAF Planning Menu

Figure 12. ATAF Offensive Counter Air Mission Input Screen

Here again, a table showing the aircraft available to fly the DCA mission type is presented to the user. In this case, though, only the aircraft to fly the DCA mission itself are entered, there are no support aircraft for this mission type. Also, the mission is untargeted. This is because DCA, DSUP, ECM, and CAP missions are flown in support of the entire ATAF, not against specific targets. As a result of being untargeted, only one mission per cycle per ATAF is allowed for each of these four mission types, since that is all that is necessary. Therefore, although these missions have a line number assigned to them, there is only one mission line number available to them.

ATAF Close Air Support Mission Input. The CAS mission is a distinct category of mission in TWX. These missions are flown in support of ground units, which are identified by a Corps ID. Figure 14 shows the ATAF Close Air Support Mission Input Screen. As with the other mission input screens, aircraft which are available to perform the CAS mission are shown in a table. These missions, like the targeted missions, have a mission line number assigned to them. The user first identifies the Corps he wishes to support. This Corps ID is checked for validity. The user then enters aircraft to fly the CAS mission. As with other mission types, more than one aircraft type may be entered per mission. This screen has full editing capabilities including the ability to recall an already committed mission for editing. The commit key (F5) and return key (F10) function as with the other screens. For that matter, the function of all of the function keys was kept consistent over the entire interface. This allows the user to more quickly become familiar with the system.



Day 1
DAY Cycle

ZATAF Defensive Counter Air Mission Input

Mission Line Number: 1401

Mission Aircraft

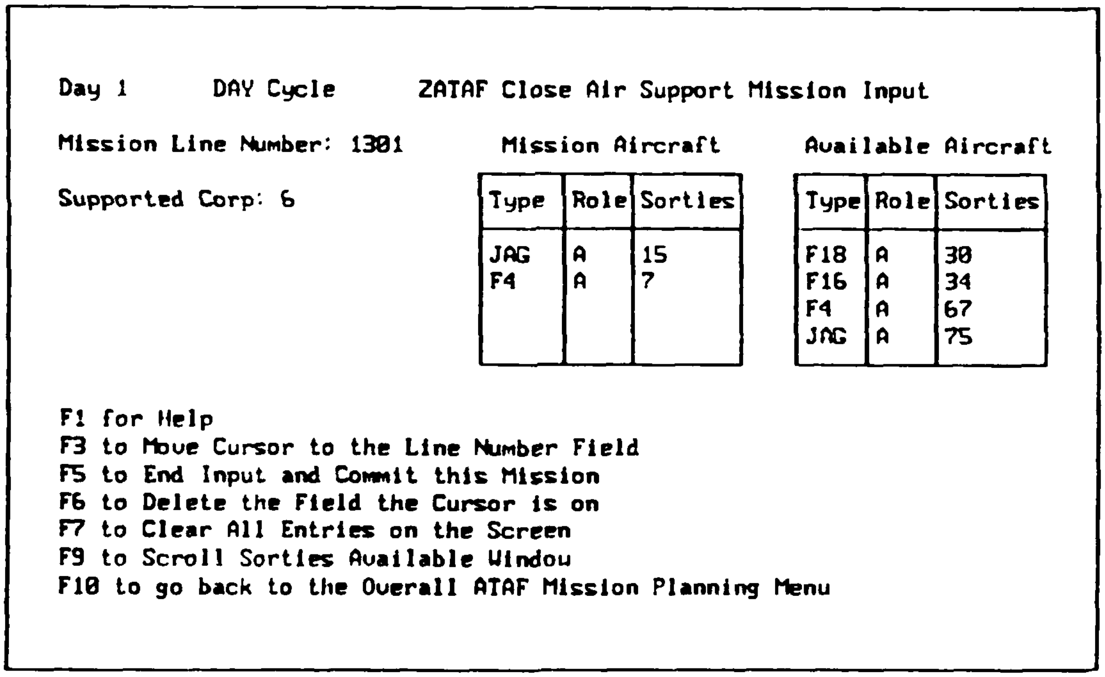
Type	Role	Sorties
F16	D	3
F4	D	39

Available Aircraft

Type	Role	Sorties
F16	D	23
F4	D	49

F1 for Help
F5 to End Input and Commit this Mission
F6 to Delete the Field the Cursor is On
F7 to Clear All Entries on the Screen
F9 to Scroll the Sorties Available Window
F10 to Return to the Overall ATAF Mission Planning Menu

Figure 13. ATAF Defensive Counter Air Mission Input Screen



Day 1 DAY Cycle ZATAF Close Air Support Mission Input

Mission Line Number: 1301

Supported Corp: 6

Mission Aircraft

Type	Role	Sorties
JAG	A	15
F4	A	7

Auailable Aircraft

Type	Role	Sorties
F18	A	30
F16	A	34
F4	A	67
JAG	A	75

F1 for Help
F3 to Moue Cursor to the Line Number Field
F5 to End Input and Commit this Mission
F6 to Delete the Field the Cursor is on
F7 to Clear All Entries on the Screen
F9 to Scroll Sorties Auailable Windou
F10 to go back to the Ouerall ATAF Mission Planning Menu

Figure 14. ATAF Close Air Support Mission Input Screen

V. Special Development and Implementation Considerations

The INGRES Development Environment

At this point, a few definitions associated with INGRES, SQL, and the relational model will help in understanding the sections which follow. These definitions will be familiar to those who understand the relational database model, SQL, and programming.

The relational model, as explained before, is a model used for database design which reduces redundancy and facilitates the logical organization of data. The relational model breaks the database up into many "tables", each of which contains related data. Each element of this data is referred to as an "attribute". All data in the database is stored in one or more tables. Of course, the INGRES DBMS, being based on the relational model, uses tables as well. An example of a table used in TWX would be an Aircraft on Airbase Table which has as attributes: Airbase ID, Aircraft Type, Aircraft Role, and Number of Aircraft.

As stated earlier, SQL (Structured Query Language) is a standard query language used for relational databases. SQL is English-like and relatively easy to learn. Statements in SQL do not tell how to retrieve data from the database, just what data to retrieve. When SQL is used with a DBMS such as INGRES, the DBMS is responsible for the retrieval and update of database tables. The DBMS user or program only needs to specify what is to be done with the data. Korth and Silberschatz provide excellent coverage of the relational model and SQL [11].

The INGRES programming language is referred to as INGRES 4GL (4th Generation Language). SQL statements are "embedded" in the INGRES 4GL. Embedding here means that the SQL statements are in the standard SQL format, but are placed in the INGRES 4GL code as needed. These embedded statements are responsible for all database retrievals and updates, and sometimes become very large and complicated.

When specifying data from tables using SQL, sometimes data from multiple tables is needed. When this is the case, multiple tables can be "joined" on common attributes. Since this is not a treatise on SQL, it should suffice here to say that these joins may become very complicated depending on the data desired. Obviously, when using a commercial SQL- based DBMS such as INGRES, memory is set aside to process these joins. A large amount of memory may be needed when processing a complicated multi-table join.

With the INGRES product, individual computer programs are used to drive the processing involved with each individual screen. A program here refers to a single application software module that can fully function alone. A screen, as before, is a set of data which appears on a computer terminal screen at the same time.

The INGRES environment allows the software associated with several screens to be grouped together when compiled and readied for execution. Such a set of software is referred to as an "application" by INGRES. Like individual modules, applications can call other applications.

Coding Documentation

During coding, documentation embedded in the source code was used. This documentation will especially help system maintainers to evaluate the effects of the various SQL statements which are embedded in the INGRES 4GL.

Likewise, each individual module was documented using the AFIT standard module header [9]. This documentation will allow the system maintainers to know the overall purpose of each module. Especially noteworthy here is that this documentation explicitly states calling and called modules. This documents the calling hierarchy of modules.

Blue and Red User Interface Considerations

Interestingly, one interface cannot serve for both the red and blue players. This is despite the fact that the two interfaces are largely identical. This two interface requirement is necessitated by a decision to horizontally partition the database by player side. Therefore, the table names used in the user interface software must be different for the red and blue sides. The blue interface was implemented first. The red interface will later be produced, using the blue screens and interface code as a template for development. A few of the basic aspects of the interfaces must be different for the red side, but they should not present a problem during implementation of the red interface.

Program Decomposition

The INGRES application development system had problems with the use of computer memory. These problems were of two basic types:

1. Some individual programs became so large that compiling them gave an INGRES "out of memory" error.
2. Some individual applications became so large that running them caused INGRES to have an "out of memory" error. Both of these problems were significant and would have stopped the development effort completely had solutions not been found to them.

The first problem, that of individual programs growing too large, was interesting in that it gave a clue to the type of program compilation that was occurring with the INGRES system. From experience, it seemed that program size was tied more heavily to the number of embedded SQL statements in the 4GL code than to any other aspect of the program. As might be expected, large multi-table joins caused the programs to be larger than simple one table queries. Obviously, the INGRES compiler was building an individual query for each embedded SQL statement in the program.

The only solution to the first problem was far from elegant. The solution was simply to get rid of any database queries that were not absolutely essential to the function of the screen and program. This was the only method that could be found. The INGRES documentation and technical support did not have another solution. In one instance, this limitation caused a screen to not have all the functions originally desired. The user was told of these limitations and instructed on ways to work around them.

The second problem was that of applications becoming too large to run in memory. The solution to this problem was to decompose the large applications into several smaller applications. Some of the applications are used to drive only one screen. Here, total system size increased, since each application carries some overhead with it. This was not a problem, though, since applications reside on the hard drive until needed, and then are loaded into memory.

Locking Considerations

The nature of the necessary PC networking, along with certain control aspects of the game, makes it necessary to implement locking procedures at both the data and application levels. When multiple users are allowed to access certain database tables at the same time, data locking must be used to ensure a consistent database. In addition application locking is needed to restrict user access to certain functions of the user interface software, depending on what functions other users are currently using or have used with the user interface.

An example of application locking in TWX is the relationship between mission planning and aircraft rerole. In TWX, some types of aircraft may be reroled from one type of role to another. For instance, F4 aircraft may be reroled from an attack role to a defense role. Only certain aircraft with specific role designators are allowed to fly particular mission types. Once the missions are planned, it is obviously unacceptable to allow the aircraft flying those missions to be reroled. This must be handled by locking users out of the aircraft rerole function of the user interface after the mission planning function has been started. The method used to implement application locking is the use of flags stored in a seminar control table in the database. Further, the consistency of these flags is assured by data level locking, which is a function available through the INGRES software.

SQL Considerations

An interesting example of the type of problems which arise during software development projects concerns the limitations of INGRES SQL. In SQL, an "outer join" is a multi-table join in which some data associated with one of the tables is desired even if there is no matching data from some other table. There is no method for specifying outer joins with the INGRES SQL. This posed a problem during development of many of the screens.

The solution to this problem was to use the INGRES screens to hold data temporarily, and then make subsequent SQL retrievals based on the values of data on the screen. For instance, suppose we desired to know the amount of each munition type at some airbase. Some of these amounts may be equal to zero. One aspect of the relational model used in the TWX database is that zero values are not recorded at all. As a result, munitions which have zero amounts at some particular airbase are not entered into any tables in the database. In this case, our solution is implemented in the manner described earlier. First, all the munition names are retrieved from a separate table in the database. Then, one munition at a time, the values are queried from the database where the munition is the one named. If there is no match, then the munition quantity will default to zero, as we would want.

VI. Testing and Validation

The final task in the user interface design effort is identical to the final tasks in designing the new database system and rehosting the simulation programs. This is the task of total system integration, validation, and testing. For the user interfaces themselves, the main testing criteria was Wargaming Center acceptance, and the verification of transactions to the database. The Wargaming Center used the prototype to determine if the user interfaces are indeed correct and useful in terms of ease of use and the exercise's information requirements. By verifying transactions, we mean that it was necessary to run several checks to make sure that the input through the user interfaces has the correct effect on corresponding database entries. For example, it was verified that sorties entered against a target through the user interface actually got scheduled against the target, as shown from entries in the database.

During the development of the new TWX user interfaces, the main standard used was Air Force Wargaming Center approval. The design included the prototype mentioned earlier. This prototype will be reviewed by Wargaming Center personnel. All suggestions that these personnel made were incorporated into the design of the interface or noted in the hook book mentioned earlier.

Due to time constraints, extensive testing of the system was not possible as part of this effort. The burden of thorough testing of the interface and the overall system as well will therefore fall on the Wargaming Center. By adhering to good software engineering principles throughout the design process, any changes required later should be easily made.

As stated earlier, the internal logic of the TWX simulations themselves remained the same. To insure that this has been accomplished, comparison testing of the new system and the old must be done. The variability in these tests may be reduced by using identical random number streams for the two simulation runs, if possible. By combining identical random number streams with identical input to both systems, a valid comparison may be made. Similar trends, at least, should appear in the results of the two simulations. This must be the standard used since the original TWX simulations are not based on any specific theoretical model.

At this point in time, only a small amount of testing comparing the results of the new and old system's simulations has been done. These results were reviewed by Wargaming Center personnel, and areas where potential discrepancies existed were noted. These areas were reviewed and corrected as needed [5].

VII. Conclusion

Review of Steps Taken

This thesis effort has attempted to solve some of the problems with the Theater War Exercise, specifically in the area of improving the TWX user interface. A solution was proposed that combined the rehosting of TWX with a total database and user interface redesign. Hardware and software, including a vendor-supplied DBMS which included an applications development package, were selected for this effort. The relational model was chosen as the basis for the new database design. Structured Query Language (SQL) was chosen as the database query language to use in the development of the user interfaces.

A background of TWX and of user interface design in general was reviewed. The literature in the area of human factors, user psychology, and cognitive models helped in understanding the human issues involved in the TWX user interface design. A design methodology based on the characteristics of the TWX system was derived from the models and methodologies presented.

Requirements gathering, design and implementation were done using the methodology derived. Various peculiarities such as the limitations of SQL, INGRES memory limitations, and the requirement for two interfaces arose during development. These were typical examples of the types of problems which occur during software development, and were addressed as well as possible.

Testing and validation considerations were a recognized part of this effort. Testing of database transactions was done as each program module for the user interface was developed. Thorough testing of the completely integrated system, including the simulations is a responsibility that must fall to the Air Force Wargaming Center.

Lessons Learned

This thesis effort was valuable in that it pointed out many of the types of successes and failures that occur in almost all software development efforts. As far as strategies which were part of the project which helped it to achieve some success, there are three in particular that are noteworthy. They are: the use of an applications development tool, the screen design methodology used, and the avoidance of referring to the current system's code.

The INGRES 4GL and screen generator tools were extremely valuable in achieving a high level of productivity in terms of screens generated in a short period of time. For any system that is user interface intensive, the use of such a tool is highly recommended. As noted before, the INGRES system had memory limitations when attempting to process large numbers of complicated queries. We anticipate that later versions of this product will correct this deficiency.

The methodology of reviewing preliminary screen designs with the Wargaming Center before coding began was also highly valuable in achieving the type of productivity that was made with this project. Once again, with a user interface intensive project, this is a recommended step for whatever overall design method is used.

A final strategy that helped with the success of the project was that of developing the system with little reference to the old system's source code. This helped in developing a logical structure of the screens and the processing that is associated with each screen. Only with certain calculations, such as those found with logistics movement tonnages, was the original source code referred to.

As far as the aspects of this effort that could have been handled differently to improve the project, two come to mind. The first is that the INGRES tool had certain limitations that were unknown at the beginning of the project which subsequently affected productivity. The second is that the representation of system requirements was lacking in one important regard.

INGRES was perhaps not as "proven" a performer as would be desired. As mentioned before the memory limitations posed problems which were difficult and awkward to work around. It is not known whether any product on the market could have handled the type and amount of processing that was necessary with the TWX user interfaces. However, if it was known that the INGRES system could not, then other products would have been considered.

The pictorial and textual representation of requirements for each screen was useful for firmly defining what processing was needed for the database updates and screen processing. However, another textual section describing and defining some of the algorithms and formulas involved in the processing behind each screen was also needed. In some cases, these requirements were really not established when implementation began, simply because certain database design and simulation issues were unresolved. However, in cases where the nature of the processing was known, having it documented would have greatly sped up the coding process.

Opportunities for Further Research

There are three areas for further research relating to this thesis effort that are of interest to the Air Force Wargaming Center and that would prove challenging. The three areas are: validation of the simulation models against standard combat models, the implementation of a land battle user interface, and the development of graphics output capabilities for TWX.

Both the air and land battle simulations need to be validated against existing combat models. Such models are plentiful in literature. Currently, the TWX simulations are not based on any model but may still reflect characteristics of some model. If it is found that this is not the case, then the simulations should be rewritten (based on some model) and other parts of the game adjusted. If instead it is found that the TWX simulations are loosely based on some model, then the simulation should be documented as such. This area of research should be especially appealing to strategic and tactical sciences students.

Development of a land interface is a high-priority item of interest to the Wargaming Center. As explained earlier, the land battle is loaded up at the beginning of each exercise day and let to run on its own. If a land interface could be developed which allows the TWX players to control the land battle, then the benefits derived from playing TWX would certainly increase. Unlike the user interfaces developed by this thesis effort, there is no existing interface to even use as a beginning reference point. As a result, the effort needed to develop a land interface would be greater, and even more interaction with the Wargaming Center would be needed to establish requirements.

The development of graphics capabilites for TWX output is also of interest to the Wargaming Center. Specifically, a terminal screen using graphics to represent the FEBA (Forward Edge of the Battle) is desired. This effort would involve establishing requirements for the FEBA's appearance, and then tying TWX output into a program which would create the FEBA map.

Bibliography

1. *Theater Warfare Exercise Handbook*. Air War College, AWC Maxwell AFB, Alabama, 1987.
2. Izak Benbasat et al. The influence of color and graphical information presentation in a managerial decision simulation. *Human-Computer Interaction 2*, Winter 1986.
3. Richard A. Bolt. *The Human Interface*. Lifetime Learning Publications, Belmont, CA, 1984.
4. Carl Braesicke et al. User interfaces working group 6. *ACM SIGSOFT Software Engineering Notes, 10,2*, April 1985.
5. Michael S. Brooks. *Develop a New Database and Support Software for the Theater War Exercise*. Master's thesis, Air Force Institute of Technology, 1987.
6. Gary W. Dickson et al. Understanding the effectiveness of computer graphics for decision support: a cumulative experimental approach. *Communications of the ACM 29*, January 1986.
7. Wilbert O. Galitz. *Handbook of Screen Format Design*. QED Information Sciences, Inc., Wellesley Hills, MA, 1985.
8. Jack Grimes et al. User interface design: are human factors principles used? *SIGCHI Bulletin, 17*, January 1986.
9. Thomas C. Hartrum. *AFIT/ENG Software Development Documentation Guidelines and Standards*. March 1986.
10. Edwin L. Hutchins et al. Direct manipulation interfaces. *Human-Computer Interaction, 1*, Fall 1985.
11. Henry F. Korth and Abraham Silberschatz. *Database System Concepts*. McGraw-Hill, Inc., New York, 1986.
12. William Mark. Knowledge-based user interface design. *Human-Computer Interaction, 1*, Fall 1985.
13. Thomas P. Moran. Summary of allen newell's chi '85 address, 'the prospects for science in human-computer interaction'. *SIGCHI Bulletin, 17*, July 1985.
14. Allen Newell and Stuart K. Card. The prospects for psychological science in human-computer interaction. *Human-Computer Interaction, 1*, Summer 1985.
15. Roger S. Pressman. *Software Engineering A Practitioner's Approach*. McGraw-Hill Book Company, New York, 1987.
16. Richard Rubenstein et al. *The Human Factor*. Digital Press, Burlington, MA, 1984.
17. Allan D. Saja. The cognitive model: an approach to designing the human-computer interface. *SIGCHI Bulletin, 16*, January 1985.
18. Ralph H. Sprague and Eric D. Carlson. *Building Effective Decision Support Systems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.
19. Duard S. Woffinden. *Interactive Environment for A Computer-Aided Design System*. Master's thesis, Naval Postgraduate School, 1984.
20. Stephen S. Yau and Jeffery J.P. Tsai. A survey of software design techniques. *IEEE Transactions on Software Engineering, 12*, June 1986.

Vita

Captain Mark S. Kross was born on 27 June 1961 in Ann Arbor, Michigan. He graduated from high school in Huntsville, Alabama in 1979 and attended the University of Alabama, where he received the degree of Bachelor of Arts in Mathematics in May 1983. Upon graduation, he received a commission in the USAF through the ROTC program. He began active duty in July of 1983. He served as a command and control systems interface analyst at Headquarters, Strategic Information Systems Division, Offutt AFB, Nebraska, until entering the School of Engineering, Air Force Institute of Technology, in May 1986.

Permanent address: 123 Lafayette Lane
Bellevue, Nebraska 68005

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

A179 6

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GCS/ENG/87D19

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b OFFICE SYMBOL (If applicable)
AFIT/ENG

7a NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Air Force Institute of Technology (AU)
Wright-Patterson AFB, Ohio 45433-6583

7b. ADDRESS (City, State, and ZIP Code)

8a NAME OF FUNDING / SPONSORING ORGANIZATION
Air Force Wargaming Center

8b OFFICE SYMBOL (If applicable)
AU CADRE/WG

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
Maxwell Air Force Base, Alabama 36112-5000

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
See Box 19

12. PERSONAL AUTHOR(S)
Mark Scott Kross, B.A., Capt, U.S. Air Force

13a. TYPE OF REPORT
MS Thesis

13b TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (Year, Month, Day)
1987 December

15 PAGE COUNT
75

16. SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
15	06	
12	05	

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
War games, Computer applications, User Needs, Computerized Simulation

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: DEVELOPING NEW USER INTERFACES FOR THE THEATER WAR EXERCISE

Thesis Chairman: Mark A. Roth, Captain, USAF,
Assistant Professor of Computer Systems

IAW AFR 190-1.
Lynn Wolaver 31 Dec 87
Dean for Research and Professional Development
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
Mark A. Roth, Captain, USAF

22b TELEPHONE (Include Area Code)
(513) 255-3576

22c OFFICE SYMBOL
AFIT/ENG

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Abstract

The Theater War Exercise (TWX) is a five-day, two-sided, theater level, computer assisted wargaming exercise. TWX is an exercise in theater-level airpower employment decision making. It is played at the Air War College and at intermediate service schools in England and Canada. It is maintained by the Air Force Wargaming Center.

This thesis sought to redress two of the problems with the TWX system. The first was that due to its mainframe application, TWX was costly to use for the Canadian and British schools. The second problem was that all input and output for TWX was made over hard copy devices, making TWX difficult and frustrating to use for exercise players. To solve these problems, this thesis effort consisted of moving the entire exercise to a microcomputer configuration and designing and implementing new, screen-oriented user interfaces emphasizing ease of use and flexibility.

First, hardware and software for the development system was selected. Decisions in the hardware and software selection area were affected by the wishes of the Air Force Wargaming Center. The body of literature concerning user interface design was then reviewed. From this literature, a design methodology to be used for the TWX user interfaces was developed.

The user interfaces were then built. Certain aspects of the development effort proved especially interesting. Among these aspects were locking considerations, use of the selected software tools, and programming problems which arose during development.

Finally, areas for further research and enhancement of the TWX system were explored. Some of these areas are of interest to those in the field of strategic and tactical sciences, as well as those interested in software design.

END

DATE

3-88

DTIC